Ph.D. DISSERTATION

Density-based Representation Learning with Applications to Sentiment Analysis and Domain Adaptation

밀도표현 학습 방법론과 감성 분석, 도메인 적응에의 응용

BY

Saerom Park

FEBRUARY 2018

DEPARTMENT OF INDUSTRIAL ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Ph.D. DISSERTATION

Density-based Representation Learning with Applications to Sentiment Analysis and Domain Adaptation

밀도표현 학습 방법론과 감성 분석, 도메인 적응에의 응용

BY

Saerom Park

FEBRUARY 2018

DEPARTMENT OF INDUSTRIAL ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Abstract

As more and more raw data are created and accumulated, it becomes important to identify information from the data. In order to analyze the collected data, machine learning and deep learning models are mainly used in recent years, but the performance of these models is highly dependent on data representation. Recent works on representation learning have shown that capturing the input density can be helpful to get useful information from data. Therefore, in this dissertation we focus on density-based representation learning. In high-dimensional data, manifold assumption is one of the important concepts in representation learning because high-dimensional data are actually concentrated near the lower dimensional high density region (manifold). For unstructured data, converting to numerical vectors is necessary to apply machine learning and deep learning models. In case of text data, distributed representation learning can effectively reflect information of input data while acquiring continuous vectors of words and documents. In this dissertation, we disentangle some issues on manifold of input data and distributed representation of text data in terms of density-based representation learning.

First, we examine denoising autoencoders (DAE) from the perspective of dynamical systems when the input density is defined as a distribution on manifold. We construct a dynamic projection system associated with the score function, which can be directly obtained from an autoencoder model that is trained from a Gaussian-convoluted input data. Several analytical results for this system are proposed and applied to develop a nonlinear projection algorithm to recognize the high-density region and reduce the noises of corrupted inputs. The effectiveness of this algorithm is verified through the experiments on toy examples and real image benchmarking

datasets.

Support vector domain description model can estimate the input density from the trained kernel radius function under some mild conditions on margin and kernel parameters. We propose a novel inductive ensemble clustering method, where kernel support matching is applied to a co-association matrix that aggregates arbitrary basic partitions in order to construct a new similarity for kernel radius function. Experimental results demonstrate that the proposed method is effective with respect to clustering quality and has robustness to induce clusters of out-of-sample data. We also develop low-density regularization methods of DAE model by exploiting the energy of the trained kernel radius function. Illustrative examples show that the regularization is effective to pull up the energy outside the support.

Learning document representation is important in applying machine learning algorithms for sentiment analysis. Distributed representation learning models of words and documents, one of neural language models, have been utilized successively in many natural language processing (NLP) tasks including sentiment analysis. However, because such models learn the embeddings only with a context-based objective, it is hard for embeddings to reflect the sentiment of texts. In this research, we address this problem by introducing a semi-supervised sentiment-discriminative objective using partial sentiment information of documents. Our method not only reflects the partial sentiment information, but also preserves local structures induced from original distributed representation learning objectives by considering only sentiment relationships between neighboring documents. Using real-world datasets, the proposed method is validated by sentiment visualization and classification tasks and achieves consistently superior performance to other representation methods in both Amazon and Yelp datasets.

NLP is one of the most important application areas in domain adaptation because a property of texts highly depends on their corpus. Many domain adaptation methods

for NLP have been developed based on the numerical representation of texts instead of on textual input. Thus, we develop a distributed representation learning method of documents and words for the domain adaptation that addresses the support separation problem, wherein the supports of different domains are separable. In this study, we propose a new method based on negative sampling. The proposed method learns document embeddings by assuming that noise distribution is dependent on a domain. The proposed method can be divided into two cases according to the dependency of the noise distribution of words on domains when training word embeddings. Through experiments on Amazon reviews, we verify that the proposed methods outperform other representation methods in terms of visualization and proxy A-distance results. We also perform sentiment classification tasks to validate the effectiveness of document embeddings, and the proposed methods achieve consistently better results compared with other methods.

Recently, there are a large amount of available data that have high dimensional representation or exist in text form, so representation learning to capture manifold of high-dimensional data and to obtain numerical vectors of text that reflect the useful information is required. Therefore, our algorithms can be helpful to suffice these requirements and applied to various data analytics tasks.

Keywords: representation learning, manifold learning, denoising autoencoder, distributed representation, sentiment analysis, domain adaptation

Student Number: 2013-21068

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter introduces the motivations and background of the research and clarifies the aims of the dissertation. Finally, the organization of this dissertation is provided.

1.1 Motivation of the Dissertation

Data analytics embraces a generic process for drawing meaningful conclusions from collected data, i.e., collecting, preparing and analyzing data, and developing, testing and revising analytical models. This process is presented in Figure 1.1. For data analytics, various machine learning and deep learning models has been developed and used. These learning models desire the appropriate numerical representations because a good representation can make a subsequent learning task easier (Goodfellow et al., 2016; Bengio et al., 2013). Therefore, learning representation becomes more and more important in data analytics.

In machine learning, representation learning discovers and represents information of structured or unstructured input data. Representation learning can embody some

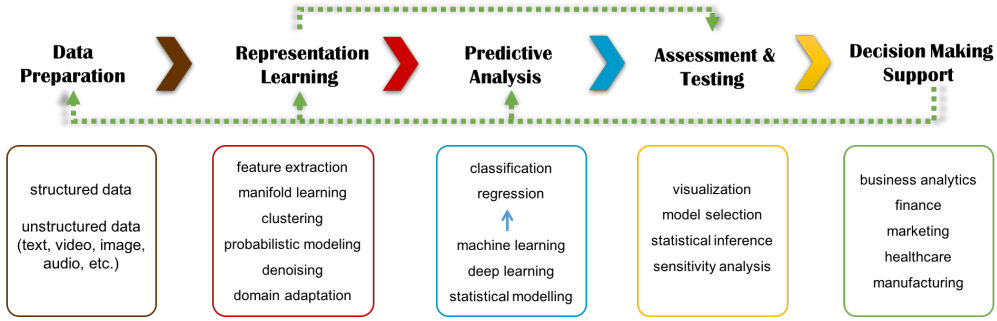


Figure 1.1: Data analytics process.

tasks such as feature extraction, manifold learning, clustering, density modeling and domain adaptation. In the past, representation learning algorithms were mainly regarded as a pre-processing step for supervised learning, but, recently, representation learning algorithms aim to find the underlying factors of variation that generates the data (Goodfellow et al., 2016). On this perspective, supervised learning can provide a direct hint as to what features are effective (Becker & Hinton, 1992; Krizhevsky et al., 2012), and unsupervised learning that tries to capture the input density can be helpful to learn a good representation that is applicable to a variety of AI tasks. Moreover, a tremendous amount of data including unstructured data have been accumulated while collecting the label information is restrictive. Unsupervised or semi-supervised learning models have been developed to learn the input density (Gopalan et al., 2014; D.-H. Lee, 2013), and transfer learning models that transfer the knowledge from well-known data set with labels to unlabeled data set have been also exploited (Pan & Yang, 2010). Learning the input density explicitly or implicitly can be applied to feature extraction, denoising, inductive clustering and domain adaptation (Vincent et al., 2010; K. Kim & Lee, 2014a; Jung et al., 2010; Gopalan et al., 2014; Jung et al., 2011; Ben-David et al., 2010).

As the collected data contains a large amount of information, data has a high-

dimensional representation. However, in case of high-dimensional data, input density concentrates on several regions (manifolds) which have even smaller dimensionality than the original data space, so manifold assumption is one of the important concepts in representation learning. Manifold learning aims to capture an effective low-dimensional structure and help in improving the performances of other machine learning algorithms. Traditional manifold learning methods (Pearson, 1901; Torgerson, 1952; Schölkopf et al., 1998; Tenenbaum et al., 2000; Roweis & Saul, 2000; Maaten & Hinton, 2008; Belkin & Niyogi, 2003) have focused on finding low-dimensional representations that preserve the original input data structure. Linear methods such as principal component analysis (PCA) (Pearson, 1901) and multidimensional scaling (MDS) (Torgerson, 1952) have been proposed, and nonlinear dimension reduction methods such as kernel PCA (Schölkopf et al., 1998), Isomap (Tenenbaum et al., 2000), locally linear embedding (LLE) (Roweis & Saul, 2000), t-sne (Maaten & Hinton, 2008), Laplacian eigenmaps (LE) (Belkin & Niyogi, 2003) and autoencoder (Hinton & Salakhutdinov, 2006a) have been proposed to handle complex nonlinear data. However, many manifold learning methods using neighbor graph are able to be easily degraded by noises and not to reconstruct points of the original input space from low dimensional representations because they learn the structure of the data-supporting manifold rather than the input density. On the other hand, an autoencoder model can learn the nonlinear mapping through the encoder function and reconstruct the inputs from the encodings through the decoder function, and regularized autoencoder models, in addition, can obtain effective features by learning a tangent plane of data manifold (Vincent et al., 2010; Rifai, Vincent, et al., 2011; Goodfellow et al., 2016) as well as be interpreted as learning the input density implicitly (Alain & Bengio, 2014; Kamyshanska & Memisevic, 2015).

Denoising autoencoder (DAE) (Vincent et al., 2010) with Gaussian noises can estimate not just the score but the Hessian of log density without any parametric

constraints on reconstruction functions (Alain & Bengio, 2014), and with symmetry of the partial derivatives of reconstruction function the potential energy that is related to negative log density with regard to Gibbs distribution can be obtained for common activation functions (Kamyshanska & Memisevic, 2015). In this respect, an autoencoder model was examined through a dynamical system in Seung (1998); Kamyshanska & Memisevic (2015). However, there was no guarantee of convergence or stability of the induced dynamical system. In addition, autoencoder models have a difficulty in capturing low-density region in the input space. On the other hand, support vector domain description (SVDD) model based on kernel method can estimate a pseudo density of the input data (D. Lee & Lee, 2007) and construct a dynamical system whose convergence to equilibrium points is guaranteed (J. Lee & Lee, 2005; K. Kim & Lee, 2014a; Jung et al., 2011). Although this model was successfully applied to clustering, classification and denoising (Jung et al., 2010; D. Lee & Lee, 2007; Jung et al., 2011; K. Kim & Lee, 2014a), the performance is highly dependent on a predefined similarity (kernel). Moreover, the induced dynamical system can be limited in estimating density because the system has equilibrium points rather than continuous ridges. Therefore, this model is deficient for finding the maximal probability ridges for the input density.

Recently, more and more information is created and stored online in the form of text from various domains. Therefore, natural language processing (NLP) application is one of the most important application fields in data analytics such as business analytics, finance and marketing, where sentiment information of texts can be popularly used. Moreover, because textual input comes from various sources, domain adaptation can be necessary to manipulate the multiple sources. However, because the textual input should be first represented as fixed-length vectors to apply density-based models and machine learning techniques, representation learning is fundamental in NLP application using machine learning models.

In NLP application, the most common approach of representing documents as vectors is the dictionary-based model using a vocabulary dictionary, that is, the bag-of-words (BOW) or bag-of-n-grams (Harris, 1954) and term-frequency inverse-document frequency (TF-IDF) (Sparck Jones, 1972). However, these models have many limits, such as ignoring the order of words in documents, obtaining high dimensional representation, and losing the semantic relationships between words (Le & Mikolov, 2014; Bengio et al., 2003; Mikolov, Chen, et al., 2013; Mikolov, Sutskever, et al., 2013). To solve these problems, many studies have been performed including using LE, LLE and autoencoder to reduce dimensionality (K. Kim & Lee, 2014b; Roweis & Saul, 2000; Zhai & Zhang, 2016; Hinton & Salakhutdinov, 2006b), projecting documents onto a subspace reflecting the semantics between words, and representing documents as mixtures of topics to deal with problems on high dimension and loss of semantic relationships (Deerwester et al., 1990; Blei et al., 2003; Perotte et al., 2011; Boyd-Graber & Resnik, 2010; Lin & He, 2009).

In recent years, many researchers have been interested in neural probabilistic language models learning a distributed representation for words and documents, which overcame the weaknesses of the dictionary-based models mentioned above (Collobert et al., 2011; Le & Mikolov, 2014; Bengio et al., 2003; Mikolov, Chen, et al., 2013; Mikolov, Sutskever, et al., 2013). Using neural networks, the distributed representation models learn continuous vectors of words and documents which preserve the similarities with respect to semantic and syntactic relations by considering the contexts, and, in a result, the input density reflecting the similarity relationships can be learned. The models can not only determine the dimension of vectors, but also exploit word order information. These models perform better in text classification and syntactic and semantic tests than other dictionary-based models (Mikolov, Sutskever, et al., 2013; Mikolov, Chen, et al., 2013; Le & Mikolov, 2014; A. M. Dai et al., 2015). However, distributed representations has trouble in disentangling synonym and antonym

relationships because they are learned based on only the context information of texts. This can be a limitation when applied to sentiment analysis.

Meanwhile, as textual inputs from various sources share similar property but have different distributions, transfer learning techniques can be usefully applied to them. When the same task such as sentiment analysis is given for textual inputs from different domains, transductive transfer learning make knowledge transfer possible. Domain adaptation is transductive transfer learning where data distributions are different according to domains in spite of single task (Pan & Yang, 2010). There are two approaches to solve the domain adaptation problem. The first approach is feature based domain adaptation. It aims to find common feature structure that can link two different domains for domain adaptation (Pan & Yang, 2010). Glorot et al. (2011) conducted a study which used stacked marginalized auto-encoder to extract common feature between different domains. Ganin et al. (2016) used the idea of adversarial training to extract the common features that cannot discriminate between the source and the target domains. Bousmalis et al. (2016) suggested domain separation networks that could learn the representation which is unique to each domain. The second one is instance-based approach. This approach focuses on revising the training of the classifier by adding various terms in loss function. There has been researches using importance weight to reweight the labeled instances from source domain (Shimodaira, 2000). Also there were researches that estimated importance weights that are used in instance based domain adaptation Bickel et al. (2009); Sugiyama et al. (2007); Gretton et al. (2009). However, these domain adaptation methods require the numerical representation as the input. Thus, text data should be first represented as a numerical vector to apply the existing domain adaptation techniques or as an domain-adapted numerical representation to apply the existing machine learning techniques.

In this dissertation, we focus on density-based representation learning, which implicitly or explicitly learns the input density using DAE and SVDD models, and rep-

resentation learning models of texual inputs for NLP applications such as sentiment analysis and domain adaptation. As explored before, density-based representation learning and its application to NLP tasks have some problems that have not yet been solved in spite of the great interest of researchers. It motivates our research to overcome the limitations of representation learning models and to obtain improved representations.

1.2 Aims of the Dissertation

This thesis aims to develop representation learning models that can well capture the input density. We concentrate on density-based representation learning based on DAE and SVDD models and distributed representation for NLP applications such as sentiment analysis and domain adaptation. In this research, we first apply our models to given data, and then assess the models by performing predictive analysis such as classification or by visualizing as in the data analytics process of Figure 1.1. The detailed objectives of this research are presented as follows:

- **Stability analysis of DAE to guarantee convergence to a manifold (Section 2)**: In density-based representation learning, a manifold is regarded as high-density region in the input space, and DAE model can implicitly learn the input density. We extend this fact to the case that an original input is generated by a *distribution on manifold* and corrupted inputs are obtained by adding Gaussian noises to the original input. From the trained reconstruction function $r(x)$ of DAE, a dynamical projection system is constructed, and convergence and stability of this system are demonstrated in terms of *distribution on manifold*. Therefore, a nonlinear projection algorithm based on this system is proposed. This algorithm can be effectively applied to denoising task, and the effect is validated through classification and visualization.

- **Inductive ensemble clustering and low-density regularization with SVDD (Section 3)**: SVDD model can estimate a pseudo density of input data that makes it possible to decompose the input space and to obtain the support region by training kernel radius function. Because kernel methods depend on a pre-defined similarity, we propose inductive ensemble clustering based on a new similarity metric by aggregating multiple clustering results. Kernel radius function can be constructed using the similarity metric, which give a robust and effective clustering results in toy and real-world datasets. We also develop low-density regularization methods for DAE model by constructing a energy from the trained kernel radius function of SVDD model. The regularization helps the induced dynamical system to capture maximal probability ridges as well as to pull up the energy outside the support. This effect is verified using illustrative examples.

- **Semi-supervised distributed representation for sentiment analysis (Section 4)**: Sentiment analysis, which analyzes opinions, sentiments, and attitudes of writers in written documents, is an important issue in NLP and has been applied to diverse fields such as business analytics, marketing, and finance (Kloptchenko et al., 2004; Gupta & Lehal, 2009; Tated & Ghonge, 2015; Rabhi et al., 2009). However, although distributed representation of words and documents can contain effective information about semantic and syntactic relationships of text, it has trouble in disentangling the sentiment information of text. To solve this problem, semi-supervised distributed representation learning method is proposed. This method aims to learn document embeddings reflecting the sentiment information by considering the partial sentiment label of neighboring documents. To evaluate the effectiveness of document representations, visualization and predictive classification are conducted in terms of sentiment

analysis.

- **Domain adapted distributed representation of words and documents (Section 5)**: Domain adaptation assumes that the distributions of training (source) and test (target) data are similar but different, unlike the assumption in traditional machine learning (Quionero-Candela et al., 2009). One of the most important issues in the domain adaptation problem is obtaining good representation because the difference in distributions fundamentally originates from representations. However, because distributed representation intactly reflects the difference of distributions, the model learns inappropriate representation for domain adaptation. We develop new domain adapted distributed representation method based on negative sampling to obtain undifferentiated document embeddings across domains by introducing the domain-dependent noise distribution. Our method can be decomposed into two cases depending on whether the domain-dependent noise distribution is introduced for word embeddings. The obtained representation can be useful for other domain adaptation methods.

1.3 Organization of the Dissertation

The remainder of this dissertation is organized as follows. In chapter 2, we propose nonlinear projection algorithm through theoretical analysis of DAE to find a lower dimensional maximal density ridge (a manifold) in higher dimensional data. In chapter 3, we construct a kernel radius function using a new similarity measure in SVDD model and regularize DAE to reflect low-density region from the energy of SVDD model using the trained kernel radius function. In chapter 4 and 5, we propose improved distributed representation methods of words and documents for NLP that can

reflect sentiment information and disentangle domain separation problem. Chapter 6 concludes this dissertation along with contribution and future plan of the research.

Chapter 2

Stability Analysis of Denoising Autoencoder

2.1 Chapter Overview

In high-dimensional representation space, the input data are often concentrated on the lower-dimensional manifold. Learning data manifold plays an important role in improving the performance of machine learning models. Traditional manifold learning methods such as PCA, MDS, Isomap, LLE and LE obtain new low-dimensional representation on linear or nonlinear subspaces deterministically, and, as a result, these methods can be easily degraded by noises and cannot reconstruct points of the original input space from low-dimensional representations (Pearson, 1901; Torgerson, 1952; Tenenbaum et al., 2000; Roweis & Saul, 2000; Belkin & Niyogi, 2003). On the other hand, there is a probabilistic perspective that a data manifold is regarded as the high-density region, and the input data are dispersed near the manifold (Alain & Bengio, 2014; K. Kim & Lee, 2014a). On this perspective, estimating the input

density is closely related to learning a data manifold.

Nonlinear projection algorithms aim to project noised points or outliers onto a data manifold of more probable points and complement and facilitate manifold learning by using projected points with reduced noises. K. Kim & Lee (2014a) explicitly estimated a support of a data manifold (an open neighborhood of a high-density region) and projected the noisy data onto the manifold using SVDD model. This estimation could cause a projection result to transmute the original input density and distort machine learning result inside the data manifold because it uses a support estimate rather than a density estimate of a data distribution. Autoencoder models try to learn and identify the structure of a high-density region by reconstructing the inputs from the encodings through a decoder function (Hinton & Salakhutdinov, 2006a; Vincent et al., 2010; Rifai, Vincent, et al., 2011; Goodfellow et al., 2016; Alain & Bengio, 2014; Kingma & Cun, 2010; Kamyshanska & Memisevic, 2015; Swersky et al., 2011; Vincent, 2011).

In particular, DAE (Vincent et al., 2010) with Gaussian noises can estimate explicitly the score of true density and Hessian of log density without any parametric constraints on reconstruction functions (Alain & Bengio, 2014). Moreover, the explicit form of log density can be obtained for many common activation functions for an autoencoder with tied weights (Kamyshanska & Memisevic, 2015). This property enables the construction of an induced dynamical system to project a less probable point (an unstable point in terms of potential energy) nonlinearly to the more probable point (a stable point or an attractor) because the score approximates the ascending direction of log density (proportional to negative energy function on a perspective of Gibbs distribution).

In this study, we develop a theoretical analysis of DAE when the original input distribution is confined to the lower-dimensional manifold, thereby extending several previous results of the DAE with the condition that an input density should be posi-

tive everywhere (Alain & Bengio, 2014; Bojchevski, 2017). We regard learning a data manifold as learning a vector field of the energy structure of the corrupted input space and construct a dynamical projection system (DPS) from an estimated score function. We first demonstrate that the equilibrium manifold of the DPS approximates the true data manifold. We then prove that the corrupted input data point asymptotically converges to a point on the equilibrium manifold of the DPS under certain mild conditions. The results of this analysis apply to developing a nonlinear projection method using DAE that projects corrupted data onto the maximal probability ridge that represents the low-dimensional manifold. We demonstrate through experiments that our projection algorithm can effectively reduce noises and validate that the projection can improve the dimension reduction performance in toy examples and classification performance in real image data.

2.2 Motivation for Using Dynamical System

Autoencoders consist of encoder and decoder functions. The encoder function h maps input data to a hidden representation, while the decoder function g reconstructs the input from the hidden representation. The composition of h and g is the reconstruction function r with $r(\mathbf{x}) = g(h(\mathbf{x}))$ that maps the original input to the predicted input, thereby minimizing mean squared error (MSE) $\|\mathbf{x}-r(\mathbf{x})\|^2$. The DAE model is trained from the noised input, $\tilde{\mathbf{x}} = \mathbf{x} + \boldsymbol{\epsilon}$, to reconstruct the original input x by minimizing $\|\mathbf{x}-r(\tilde{\mathbf{x}})\|^2$. In Alain & Bengio (2014), $r(\mathbf{x})-\mathbf{x}$ can provide us with an estimator of the score when an over-complete autoencoder is regularized with denoising or contractive penalties.

$$r(\mathbf{x}) - \mathbf{x} = \sigma^2 \frac{\partial \log p_x(\mathbf{x})}{\partial x} + o(\sigma^2), \quad \sigma^2 \to 0 \tag{2.1}$$

which is valid only in the small noise variance, where $p_\epsilon(\boldsymbol{\epsilon}) = \mathcal{N}(0, \sigma^2\mathbf{I})$, and $p_x \in \mathcal{C}^1$ is the probability density with $p(\mathbf{x}) \neq 0$ for all $\mathbf{x}$. equation (2.1) implies that $\|r(\mathbf{x}) - \mathbf{x}\|$

is small at the maximum of the probability, and the vector field $r(\mathbf{x}) - \mathbf{x}$ moves towards the maximum probability ridge (Kamyshanska & Memisevic, 2015).

This study aims to provide a novel projection method that can project the noisy data (nonlinearly) onto a low-dimensional subspace determined by the trained DAE, as illustrated in Figure 2.1, so as to stably search for a low-dimensional structure using the projected data.

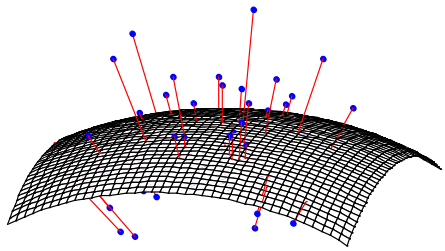


Figure 2.1: Projection on nonlinear manifold using the dynamical system.

In the case of linear autoencoder or equivalently principal component analysis (PCA), it is easy to project the noisy data (linearly) onto a low-dimensional space called a principal plane spanned by principal components (Figure 2.2). Specifically, let the linear encoder be $h(\mathbf{x}) = V_h^T \mathbf{x}$ for a centered dataset $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$ and the linear decoder be $g(\mathbf{h}) = V_g \mathbf{h}$, where $V_h, V_g \in \mathbb{R}^{n \times m}$. Then, the reconstruction function r, which minimizes reconstruction error, becomes (Goodfellow et al., 2016)

$$r(\mathbf{x}) = VV^T \mathbf{x} \tag{2.2}$$

where $V_h = V_g = V = [v_1 \ \cdots \ v_m] \in \mathbb{R}^{n \times m}$ and $v_1, ..., v_m$ are the orthogonal eigenvectors of $C = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i \mathbf{x}_i^T$, called the principal components. If we let $\mathcal{M}$ be the principal plane spanned by V, then the projected point onto $\mathcal{M}$ for a given noisy data point $\mathbf{x}_0$, that is, $\mathbf{z}_0$, is given by

$$\mathbf{z}_0 = VV^T \mathbf{x}_0 \tag{2.3}$$

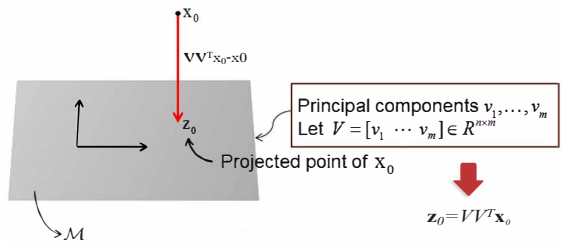


Figure 2.2: Projection on the principal plane by linear autoencoder.

In a linear autoencoder model, the dynamical system in the input space can be naturally defined because the projected point can be obtained using the following linear system:

$$\frac{d\mathbf{x}(t)}{dt} = VV^T\mathbf{x}(t) - \mathbf{x}(t) \tag{2.4}$$

where $\mathbf{x}(0) = \mathbf{x}_0 = \mathbf{z}_0 + \mathbf{y}_0$.

We obtain the following equation because we can decompose $\mathbf{x}(t) = \mathbf{z}(t) + \mathbf{y}(t)$, where $\mathbf{z}(t) \in \mathcal{M}$ and $\mathbf{y}(t) \in \mathcal{M}^\perp$ (orthogonal complement of $\mathcal{M}$), for all $t \in \mathbb{R}$, by the well-known projection theorem:

$$\frac{d\mathbf{x}(t)}{dt} = \frac{d(\mathbf{z}(t) + \mathbf{y}(t))}{dt} = (VV^T - I)(\mathbf{z}(t) + \mathbf{y}(t)). \tag{2.5}$$

We arrive at the following two separate equations:

- Since $(VV^T - I)\mathbf{z}(t) = 0$ for $\forall \mathbf{z}(t) \in \mathcal{M}$, $\frac{d}{dt}\mathbf{z}(t) = 0$, which implies $\mathbf{z}(t) = \mathbf{z}_0$ for all t.
- $\frac{d}{dt}\mathbf{y}(t) = (VV^T - I)\mathbf{y}(t)$ implies $\mathbf{y}(t) = e^{-[I-(VV^T)]t}\mathbf{y}_0 \to 0$ as $t \to \infty$

Therefore, applying dynamical system (2.4) provides $\mathbf{x}(t) \to \mathbf{z}_0$ as $t \to \infty$, which is the same result as using the PCA in equation (2.3). This scenario motivates the use of the dynamical system in the case of linear projection.

In the case of nonlinear projection onto a nonlinear manifold $\mathcal{M} \subset \mathbb{R}^n$, the following dynamical system which generalizes the linear system (2.4)

$$\frac{d\mathbf{x}(t)}{dt} = r(\mathbf{x}(t)) - \mathbf{x}(t) \tag{2.6}$$

can be applied to a noisy data point $\mathbf{x}_0$ from which we can obtain the projected point $\mathbf{z}_0$. The projection on nonlinear manifold using the dynamical system is illustrated in Figure 2.1.

2.3 Stability Analysis of the Dynamical Projection System

In this study, we assume that the original input data lie on a low-dimensional smooth-connected manifold $\mathcal{M}$ embedded in a high-dimensional space $\mathbb{R}^n$, but are corrupted by noises near manifold $\mathcal{M}$. In the case of continuous input data, the DAE learns to represent a probability distribution of input data implicitly, where an explicit $p_{model}(\mathbf{x}; \theta)$ is difficult to obtain, but the score of the distribution can be estimated. The proposed method from the estimated score, which is the log gradient of the blurred distribution, projects the input points onto a manifold using a dynamical system because the projection only requires a gradient of function for minimization or maximization. Our algorithm determines a trajectory that minimizes the energy function. Furthermore, adding the Gaussian noise to the original training data is helpful in smoothing the distribution of finite training data and overcoming the local minima problem, as mentioned in Goodfellow et al. (2016); Kingma & Cun (2010). Therefore, we will show that the DAE with Gaussian noises learns the score of the corrupted input distribution without any constraints on the functional form of the DAE model, and the obtained manifold from the score can finally approximate the true manifold.

Let $\mathbf{x} \in \mathcal{M} \subset \mathbb{R}^n$ denote a random original input vector, and $\tilde{\mathbf{x}} \in \mathbb{R}^n$ denotes a

random corrupted input vector with joint distribution $p_{x,\tilde{x}}(\mathbf{x}, \tilde{\mathbf{x}})$. The original input vector is assumed to be generated by a *distribution on manifold* $\mathcal{M}$, that is, the distribution is confined in the manifold $\mathcal{M}$ where the marginal distribution $p_x(\mathbf{x}) > 0$ on $\mathbf{x} \in \mathcal{M}$, but $p_x(\mathbf{x}) = 0$ on $\mathbf{x} \notin \mathcal{M}$. The objective of the DAE with a squared loss function is to minimize MSE defined by

$$\text{MSE}(r) := \mathbb{E}[\|\mathbf{x} - r(\tilde{\mathbf{x}})\|^2] = \mathbb{E}_{\tilde{\mathbf{x}}}\mathbb{E}_{\mathbf{x}|\tilde{\mathbf{x}}}[\|\mathbf{x} - r(\tilde{\mathbf{x}})\|^2|\tilde{\mathbf{x}}] \tag{2.7}$$

and can be solved by minimizing $\text{MSE}(r)$ point-wise as

$$\begin{aligned} r(\tilde{\mathbf{x}}) &= \arg\min_{\mathbf{a}} \mathbb{E}_{\mathbf{x}|\tilde{\mathbf{x}}}[\|\mathbf{x} - \mathbf{a}\|^2|\tilde{\mathbf{x}}] \\ &= \arg\min_{\mathbf{a}} \left(\mathbb{E}_{\mathbf{x}|\tilde{\mathbf{x}}}[\|\mathbf{x}\|^2|\tilde{\mathbf{x}}] - 2\mathbf{a}^T\mathbb{E}_{\mathbf{x}|\tilde{\mathbf{x}}}[\mathbf{x}|\tilde{\mathbf{x}}] + \|\mathbf{a}\|^2\right) \end{aligned}$$

The solution $r(\tilde{\mathbf{x}})$ is then

$$r(\tilde{\mathbf{x}}) = \mathbb{E}^{\mathcal{M}}_{\mathbf{x}|\tilde{\mathbf{x}}}[\mathbf{x}|\tilde{\mathbf{x}}] = \int_{\mathcal{M}} \mathbf{x} p_{x|\tilde{x}}(\mathbf{x}|\tilde{\mathbf{x}}) d\mathbf{x} \tag{2.8}$$

To the best of our knowledge, this result for *distribution on manifold* has not been presented in the literature previously and corrects and extends the result of Appendix A in Alain & Bengio (2014).

Assume that the corrupted input vector satisfies $\tilde{\mathbf{x}} = \mathbf{x} + \boldsymbol{\epsilon}$ with the conditional Gaussian distribution:

$$p_{\tilde{x}|x}(\tilde{\mathbf{x}}|\mathbf{x}) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{\|\tilde{\mathbf{x}} - \mathbf{x}\|^2}{2\sigma^2}\right) \tag{2.9}$$

Then, the result obtained in Bojchevski (2017) can be similarly derived and extended to a *distribution on manifold* $\mathcal{M}$ using the techniques of integration on man-

ifold as follows:

$$
\begin{aligned}
&\bigtriangledown_{\tilde{\mathbf{x}}} \log p_{\tilde{x}}(\tilde{\mathbf{x}}) \\
&= \frac{\bigtriangledown_{\tilde{\mathbf{x}}} p_{\tilde{x}}(\tilde{\mathbf{x}})}{p_{\tilde{x}}(\tilde{\mathbf{x}})} = \frac{\bigtriangledown_{\tilde{\mathbf{x}}} \int_{\mathcal{M}} p_{\tilde{x}|x}(\tilde{\mathbf{x}}|\mathbf{x}) p_x(\mathbf{x}) d\mathbf{x}}{p_{\tilde{x}}(\tilde{\mathbf{x}})} \\
&= \frac{\int_{\mathcal{M}} \bigtriangledown_{\tilde{\mathbf{x}}} p_{\tilde{x}|x}(\tilde{\mathbf{x}}|\mathbf{x}) p_x(\mathbf{x}) d\mathbf{x}}{p_{\tilde{x}}(\tilde{\mathbf{x}})} \\
&= \int_{\mathcal{M}} \frac{\mathbf{x} - \tilde{\mathbf{x}}}{\sigma^2} \frac{p_{\tilde{x}|x}(\tilde{\mathbf{x}}|\mathbf{x}) p_x(\mathbf{x})}{p_{\tilde{x}}(\tilde{\mathbf{x}})} d\mathbf{x} \\
&= \int_{\mathcal{M}} \frac{\mathbf{x} - \tilde{\mathbf{x}}}{\sigma^2} \frac{p_{x,\tilde{x}}(\mathbf{x}, \tilde{\mathbf{x}})}{p_{\tilde{x}}(\tilde{\mathbf{x}})} d\mathbf{x} \\
&= \frac{1}{\sigma^2} \int_{\mathcal{M}} (\mathbf{x} - \tilde{\mathbf{x}}) p_{x|\tilde{x}}(\mathbf{x}|\tilde{\mathbf{x}}) d\mathbf{x} \\
&= \frac{1}{\sigma^2} \left(\int_{\mathcal{M}} \mathbf{x} p_{x|\tilde{x}}(\mathbf{x}|\tilde{\mathbf{x}}) d\mathbf{x} - \int_{\mathcal{M}} \tilde{\mathbf{x}} p_{x|\tilde{x}}(\mathbf{x}|\tilde{\mathbf{x}}) d\mathbf{x} \right) \\
&= \frac{1}{\sigma^2} \left(\mathbb{E}^{\mathcal{M}}_{\mathbf{x}|\tilde{\mathbf{x}}}[\mathbf{x}|\tilde{\mathbf{x}}] - \tilde{\mathbf{x}} \right) = \frac{1}{\sigma^2} \left(r(\tilde{\mathbf{x}}) - \tilde{\mathbf{x}} \right)
\end{aligned}
$$

The above equation shows that the gradient of the log probability with respect to the corrupted input space, that is, the score function, is proportional to the difference between the corrupted input and its optimal reconstructed value using the DAE. The DAE with sufficient overcomplete hidden units has proven to be sufficiently capable of learning a complex input density function and hence can provide a favorable approximation to this result although it can be influenced by the training process in practice.

One prominent feature of the reconstruction function r is that the high-density region in the corrupted input space approaches the true manifold by reducing the noise level, as presented below.

Theorem 1. *Let $r : \mathbb{R}^n \to \mathbb{R}^n$ be a Lipschitz continuous function that solves equation (2.7) when the original input vectors are generated by the distribution $p_x(\mathbf{x})$ on manifold $\mathcal{M}$, i.e., $p_x(\mathbf{x}) > 0$ on $\mathbf{x} \in \mathcal{M}$, but $p_x(\mathbf{x}) = 0$ on $\mathbf{x} \notin \mathcal{M}$. If we let $\mathcal{M}_{\sigma^2} = \{\tilde{\mathbf{x}} \in \mathbb{R}^n : r(\tilde{\mathbf{x}}) = \tilde{\mathbf{x}}\}$, then $\mathcal{M}_{\sigma^2}$ is the high-density region in the corrupted*

input space and converges to $\mathcal{M}$ *as* $\sigma^2 \to 0$.

Proof: First of all, $\mathcal{M}_{\sigma^2} = \{\tilde{\mathbf{x}} \in \mathbb{R}^n : r(\tilde{\mathbf{x}}) = \tilde{\mathbf{x}}\}$ is an equilibrium manifold of system (2.10) by definition. Secondly, we obtain $p_{\tilde{x}|x}(\tilde{\mathbf{x}}|\mathbf{x}) \to \delta(\tilde{\mathbf{x}} - \mathbf{x})$ where $\delta(\tilde{\mathbf{x}}, \mathbf{x})$ is the delta function, and $p_{\tilde{x}}(\tilde{\mathbf{x}}) \to p_x(\tilde{\mathbf{x}})$, when $\sigma^2 \to 0$. Therefore, if $\tilde{\mathbf{x}}$ is obtained from $p_{\tilde{x}}(\tilde{\mathbf{x}})$ and $\tilde{\mathbf{x}} \in \mathcal{M}$, then as $\sigma^2 \to 0$,

$$\begin{aligned} r(\tilde{\mathbf{x}}) &= \int_{\mathcal{M}} \mathbf{x} p_{x|\tilde{x}}(\mathbf{x}|\tilde{\mathbf{x}}) d\mathbf{x} = \int_{\mathcal{M}} \mathbf{x} \frac{p_{\tilde{x}|x}(\tilde{\mathbf{x}}|\mathbf{x}) p_x(\mathbf{x})}{p_{\tilde{x}}(\tilde{\mathbf{x}})} d\mathbf{x} \\ &\to \frac{1}{p_x(\tilde{\mathbf{x}})} \int_{\mathcal{M}} \mathbf{x} \delta(\tilde{\mathbf{x}} - \mathbf{x}) p_x(\mathbf{x}) d\mathbf{x} = \frac{\tilde{\mathbf{x}} p_x(\tilde{\mathbf{x}})}{p_x(\tilde{\mathbf{x}})} = \tilde{\mathbf{x}}, \end{aligned}$$

where the shifting property of delta function is used (Bracewell, 1999).

Otherwise, i.e. if $\tilde{\mathbf{x}} \notin \mathcal{M}$, then $\int_{\mathcal{M}} \mathbf{x} \delta(\tilde{\mathbf{x}}, \mathbf{x}) p_x(\mathbf{x}) d\mathbf{x} = 0$. Thus, we obtain $r(\tilde{\mathbf{x}}) \to 0$ as $\sigma^2 \to 0$, which is intuitively obvious from $P(\tilde{\mathbf{x}} \in \mathcal{M}) \to 0$. Therefore $\mathcal{M}_{\sigma^2}$ converges to $\mathcal{M}$ as $\sigma^2 \to 0$. This result represents the perfect reconstruction of a manifold $\mathcal{M}$ and generalizes that of the proof in Appendix C of (Alain & Bengio, 2014) that assumes $p_x(\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathbb{R}^n$. ¶

This result implies that identifying a high-density region in the corrupted input space is equivalent to finding the data manifold. Therefore, we can project a corrupted input data onto the manifold nonlinearly but orthogonally by increasing the log probability, $\log p_{\tilde{x}}(\tilde{\mathbf{x}})$, of the corrupted input space in the direction of its score function. This projection motivates the use of the following *dynamical projection system* associated with reconstruction function r learned from corrupted data:

$$\frac{d\tilde{\mathbf{x}}}{dt} = \bigtriangledown_{\tilde{\mathbf{x}}} \log p_{\tilde{x}}(\tilde{\mathbf{x}}) = r(\tilde{\mathbf{x}}) - \tilde{\mathbf{x}}. \tag{2.10}$$

The (local) existence of a unique solution (or trajectory) $\tilde{\mathbf{x}}(\cdot)$ for each initial condition $\tilde{\mathbf{x}}(0) = \tilde{\mathbf{x}}_0$ is guaranteed when r is Lipschitz continuous, which includes many types of activation functions. From system (2.10), a system can be defined by

$$\frac{d\tilde{\mathbf{x}}}{dt} = \frac{r(\tilde{\mathbf{x}}) - \tilde{\mathbf{x}}}{1 + \|r(\tilde{\mathbf{x}}) - \tilde{\mathbf{x}}\|}. \tag{2.11}$$

The system (2.11) is complete, and both systems (2.10) and (2.11) are topologically equivalent. We assume without loss of generality that system (2.10) is complete (i.e., $\tilde{\mathbf{x}}(\cdot)$ is defined on all $t \in \mathbb{R}$ for any initial $\tilde{\mathbf{x}}_0 \in \mathbb{R}^n$). A state vector $\tilde{\mathbf{x}}$ satisfying the equation $\frac{d\tilde{\mathbf{x}}}{dt} = 0$ is called an equilibrium point of system (2.10). A connected component consisting of non-isolated equilibrium points of (2.10) is called an *equilibrium manifold* of system (2.10).

Another distinguishing feature of system (2.10) is that the trajectory starting from a point in a neighborhood of $\mathcal{M}_{\sigma^2}$ converges orthogonally to a point on $\mathcal{M}_{\sigma^2}$, which is established in the next theorem.

Theorem 2. *Under the condition of Theorem 1, $\mathcal{M}_{\sigma^2}$ is the attracting (stable) equilibrium manifold of system (2.10). Moreover, suppose that for any closed subset $C \subset (\mathcal{M}^\varepsilon \setminus \mathcal{M}_{\sigma^2})$, where $\mathcal{M}^\varepsilon = \{\tilde{\mathbf{x}} : \eta - \varepsilon \leq \log p_{\tilde{x}}(\tilde{\mathbf{x}}) \leq \eta\}$, $\eta = \max_{\tilde{\mathbf{x}}} \log p_{\tilde{x}}(\tilde{\mathbf{x}})$, and $\varepsilon > 0$, we obtain $\inf\{\|r(\tilde{\mathbf{x}}) - \tilde{\mathbf{x}}\| : \tilde{\mathbf{x}} \in C\} > 0$. Then, generically, system (2.10) is completely stable, i.e. every trajectory converges to $\mathcal{M}_{\sigma^2}$. In particular, if $\mathcal{M}_{\sigma^2}$ is bounded, then system (2.10) is completely stable.*

Proof: First of all, $\mathcal{M}_{\sigma^2} = \{\tilde{\mathbf{x}} \in \mathbb{R}^n : r(\tilde{\mathbf{x}}) = \tilde{\mathbf{x}}\}$ is the unique equilibrium manifold of system (2.10) by definition. Now let $V(\tilde{\mathbf{x}}) = \log(1/p_{\tilde{x}}(\tilde{\mathbf{x}}))$. Then, V is a Lyapunov function for system (2.10) since

$$\frac{dV(\tilde{\mathbf{x}}(t))}{dt} = \bigtriangledown_{\tilde{\mathbf{x}}} V(\tilde{\mathbf{x}}(t))^T \frac{d\tilde{\mathbf{x}}(t)}{dt} = -\| \bigtriangledown_{\tilde{\mathbf{x}}} \log p_{\tilde{x}}(\tilde{\mathbf{x}}(t))\|^2 < 0,$$

which implies that the trajectory $\tilde{\mathbf{x}}(t)$ starting from $\tilde{\mathbf{x}}_0$ nearby $\mathcal{M}_{\sigma^2}$ converges to a point on $\mathcal{M}_{\sigma^2}$ and orthogonally because the direction of the trajectory is proportional to $\bigtriangledown_{\tilde{\mathbf{x}}} \log p_{\tilde{x}}(\tilde{\mathbf{x}}(t))$. Therefore, $\mathcal{M}_{\sigma^2}$ is an attracting (stable) equilibrium manifold of system (2.10).

Second, because $\mathcal{M}_{\sigma^2}$ is the set of all the equilibrium points and does not intersect C by definition, C does not contain any equilibrium point of system (2.10) and satisfies

for all $\tilde{\mathbf{x}} \in C$

$$\sup\left(\frac{-\bigtriangledown_{\tilde{\mathbf{x}}} V(\tilde{\mathbf{x}})^T (r(\tilde{\mathbf{x}}) - \tilde{\mathbf{x}})}{\|r(\tilde{\mathbf{x}}) - \tilde{\mathbf{x}}\|}\right) = -\inf\{\|r(\tilde{\mathbf{x}}) - \tilde{\mathbf{x}}\|\} < 0$$

Hence in addition to the first three assumptions (A1-A3) in J. Lee & Chiang (2002) that are satisfied generically for a complete stable system, the fourth energy function assumption (A4) holds. Therefore, generically, system (2.10) is completely stable. In particular, let $\mathcal{M}_{\sigma^2}$ be bounded. Then any closed subset $C \subset (\mathcal{M}^{\varepsilon} \setminus \mathcal{M}_{\sigma^2})$ becomes compact. So there exists $\tilde{\mathbf{x}} \in C$ that takes the infimum value of $\|r(\tilde{\mathbf{x}}) - \tilde{\mathbf{x}}\|$ over C and hence $\inf_{\tilde{\mathbf{x}} \in C} \|r(\tilde{\mathbf{x}}) - \tilde{\mathbf{x}}\| > 0$. The result then follows. ¶

This result implies that we can find a point $\mathbf{x}$ on $\mathcal{M}$, which corresponds to a noisy data point $\tilde{\mathbf{x}}$ by applying system (2.10) for a small noise variance. Figure 2.3 (c) illustrates the projection of system (2.10).

2.4 Nonlinear Projection Algorithm

We propose a nonlinear projection algorithm on the basis of the stability analysis of the dynamic projection system (2.10) associated with a trained DAE in Section 2.3, as depicted in Algorithm 1. This algorithm can project a given point onto the manifold through the trajectory that converges to M because the trajectories of system (10) asymptotically converge to M under several mild conditions.

This algorithm consists of three parts: (A1) for training the autoencoder model with Gaussian noises from the data, (A2) for constructing the dynamical system using the estimated score function from the trained DAE model, and (A3) for projecting the points to the direction of the manifold.

In *A1* part, we train the DAE model by adding the Gaussian noises to the given training data. In our experiments, we set the isotropic Gaussian for the noise distribution because our datasets, such as toy examples and image datasets, are already

Algorithm 1 Nonlinear projection algorithm using DAE model

A1 Training the DAE

Given the training data $\mathcal{D} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ on the dispersed manifold $\mathcal{M}_\varepsilon$ and the noise level σ

Make the noised data with Gaussian noises:

$\tilde{\mathbf{x}}_i = \mathbf{x}_i + \epsilon,\ \epsilon \sim \mathcal{N}(0, \sigma\mathbf{I})$.

Train the DAE model by minimizing the objective:

$\sum_{i=1}^{n} \|r(\tilde{\mathbf{x}}_i) - \mathbf{x}_i\|^2$

A2 Constructing a dynamical system using the DAE

Construct a nonlinear dynamical projection system using equation (2.10)

A3 Projecting test data onto the manifold

Given the test data $\mathcal{D}_\mathcal{T} = \{\mathbf{y}_1, ..., \mathbf{y}_m\}$, the tolerance ε and the maximum number of iteration T,

for $i = 1, ..., m$ **do**

  $\mathbf{y}^s = \mathbf{y}_i$ and $t = 0$

  while $\|r(\mathbf{y}^s) - \mathbf{y}^s\| > \varepsilon \vee t \leq T$ **do**

    Numerically integrate the system (2.10) with small step size $\delta > 0$.

    $\mathbf{y}^{s+1} = (1 - \delta)\mathbf{y}^s + \delta r(\mathbf{y}^s)$

    $\mathbf{y}^s = \mathbf{y}^{s+1}$ and $t = t + 1$

  end while

  $\hat{\mathbf{y}}_i = \mathbf{y}^s$

end for

Obtain a new projected dataset $\hat{\mathcal{D}}_\mathcal{T} = \{\hat{\mathbf{y}}_1, ..., \hat{\mathbf{y}}_m\}$ near the manifold.

normalized. However, in general, it is necessary to adjust the standard deviation σ of the noise distribution according to the input distribution after normalizing the input variables to have similar scales. The prevention of the reconstruction function from approaching the trivial identity mapping is difficult because of small noises, although small noise level can approximate our projection results well to the manifold M.

Thus, we partition data into training and validation datasets, train the DAE model with the training dataset for some noise levels, and determine the noise level that reconstructs the corrupted data well in the validation dataset. The DAE model for *A2* part is obtained by fitting the whole data from the determined noise level. In *A2*, we construct dynamical system (2.10) by using the trained reconstruction function. In *A3*, the test points are projected near the manifold by integrating the dynamical system (2.10) with small steps and following the induced trajectories. The projection is completed after the predetermined steps or reaching equilibrium region $\mathcal{M}_{\sigma^2}$ that we identify with $\|r(\mathbf{x}) - \mathbf{x}\| \leq \varepsilon$ for certain tolerance ε. This procedure is demonstrated in Algorithm 1.

In the projection phase, test data are attracted near the manifold by following the trajectory of system (2.10). We can regard *A1* and *A2* as training phases and *A3* as a test phase. Therefore, we can reduce the noises along the manifold based on our nonlinear projection algorithm, and perceiving the manifold structure becomes easier for other machine learning models than prior to projection. This fact is validated through the experiments on image data in Section 2.5.

2.5 Experimental Results

We conduct the experiments on 3D toy examples and two real-world image datasets to evaluate the proposed projection method. We follow the dynamic projection procedure in Algorithm 1 and obtain new points with reduced noises from the manifold. In

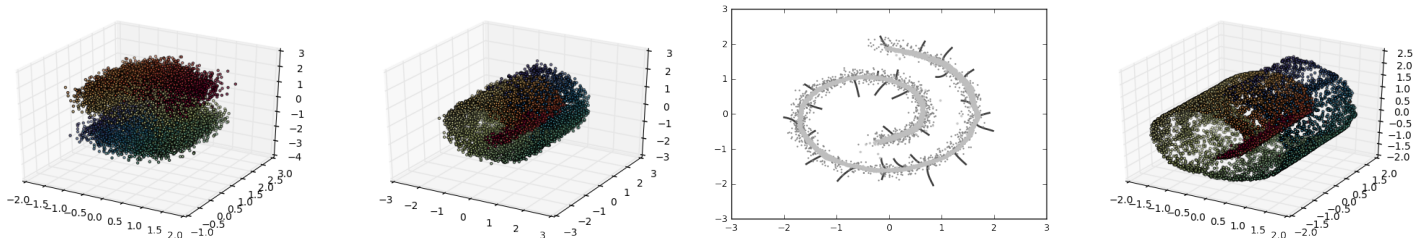
training the DAE, we use *Keras* library, which was written in Python[1] and run on GPU (Chollet, 2015). We compare our method with a nonlinear dynamic projection method, which uses the support function of SVDD model as pseudo-density function (K. Kim & Lee, 2014a). We select this SVDD-based nonlinear projection method as a comparative method because this method also uses a dynamic system and nonlinearly projected the noised data onto the manifold. This nonlinear projection method with SVDD is implemented in MATLAB (K. Kim & Lee, 2014a).

On the 3D toy example, we visualize and compare the original noised and projected data and verify the performance of the manifold in terms of capturing through projection by applying the locally linear embedding (LLE) (Roweis & Saul, 2000), which is a popular manifold learning method. We selected the LLE because of its vulnerability to noises. We use MNIST (LeCun et al., 2010) and the street view house numbers (SVHN) datasets. The SVHN dataset is a real-world dataset that includes the house numbers in Google street view images (Netzer et al., 2011). For real-world image datasets, we make clean images corrupted with noises and project the noised data through nonlinear projection algorithms from the DAE and SVDD. We evaluate the effect of reducing noises by comparing the classification performances because we cannot visualize the manifold of high-dimensional image data. Further details on experimental designs are described in the following sections.

2.5.1 Toy Examples

We generated 3D Swiss roll and S-curve datasets, which have 2D nonlinear manifolds, and made training data slightly corrupted around the manifolds, as displayed in Figure 2.3 (a) and 2.3 (b). We also made test data more corrupted than training data for testing. In the training phase, we used a one-layer over-complete autoencoder network with 1000 hidden units, where the encoder function was set to $f(\mathbf{x}) = \text{relu}(a+$

[1] https://keras.io/



(a) (b) (c) (d)

Figure 2.3: 3D toy examples: (a) corrupted S-curve data and (b) corrupted Swiss roll data, and illustration of projection procedure for Swiss roll data (c) projection trajectory of our system and (d) projected 3D plot.

$W\mathbf{x}$), and the decoder function was set to $g(\mathbf{h}) = b + V\mathbf{h}$.

Figure 2.3 (c) and 2.3 (d) depict the trajectories of our dynamical system based on the DAE and the result of the projection for test points in the Swiss roll data. In Figure 2.3 (c), we illustrated the trajectories of several corrupted test points with bold black lines in a 2D plot, where the test data were represented as gray dots, and the high-density region was light gray. The test points were projected onto the high-density region consistent with the shape of the Swiss roll manifold along the nonlinear trajectories. The test points after projection are plotted in Figure 2.3 (d) and form a 2D manifold similar to the original Swiss roll. In *A3* of our algorithm, the tolerance was introduced instead of strictly zero because we numerically integrated the dynamical system (2.10). It allowed us to adjust the level of dispersion around the manifold after projection. In Figure 2.3 (d), the connectedness of the manifold was conserved after projection since our algorithm could not capture maximum probability points but maximal probability ridge unlike (K. Kim & Lee, 2014a).

We applied LLE to the corrupted and projected test data using the SVDD and DAE to verify that the projection is effective in reducing noises toward the manifold. Figure 2.4 illustrates the dimension reduction results of the test data in the left, the projected data using SVDD in the middle, and the projected data from our algorithm in the right. We can compare the results on the basis of the preservation of the color sequence in 2D representations. Consequently, the projected data from our algorithm produced the best dimension reduction result in the Swiss roll and S-curve datasets. The test points were mixed with the non-contiguous points of the manifold, as presented in Figure 2.4 (a), and clustered with the non-adjacent points in the manifold, as displayed in Figure 2.4 (d). Although the projection using SVDD improved the dimension reduction performances, the projected points still had difficulties in unrolling the manifolds as in Figure 2.4 (b) and 2.4 (e) because the projection algorithm using SVDD could not sufficiently reduce the noises as preserving the connectedness of the

manifold. By contrast, Figure. 2.4 (c) and 2.4 (f) illustrate that the projected points from our algorithm disentangle the 2D manifold well by locating the points near the manifold. This illustration implies that our projection method can identify where the density concentrates and project the points while maintaining the manifold and its connectivity.

2.5.2 Real Datasets

We used the MNIST dataset, which consists of 28×28 gray-scale images with 10 classes (from 0 to 9), to prove the validity of our method in a high-dimensional real-world data We randomly selected MNIST images that are corrupted by salt-and-pepper noises for different proportions of noised data (10%, 30%, 50%, and 70%). For the experiment, we used 10,000 training images to train the DAE and 10,000 test images that are projected by *A3* in Algorithm 1. In the training phase, we developed and fixed a stacked autoencoder network with three relu encoding and two relu and one sigmoid decoding layers with 1000 hidden units. The network was trained with Gaussian corruption noises $\sigma = 0.2$. In the projection phase, we determined the tolerance ε as the mean of l_2-norm distances minus one-half the standard deviation, which was calculated from the estimated scores of training data.

We visualized the noised and transformed images of the MNIST dataset with 50% corruption proportion in Figure 2.5 before comparing the classification results. In addition to the projected images from the SVDD and the proposed method, we transformed the images through the trained reconstruction function, which corresponds to following the below discrete dynamical system based on the estimated score function:

$$\begin{aligned} \mathbf{y}_{t+1} &= \mathbf{y}_t + \eta \nabla_{\mathbf{x}} \log p_{\tilde{\mathbf{x}}}(\mathbf{x})|_{\mathbf{x}=\mathbf{y}_t} \\ &= \mathbf{y}_t + (r(\mathbf{y}_t) - \mathbf{y}_t) = r(\mathbf{y}_t). \end{aligned} \tag{2.12}$$

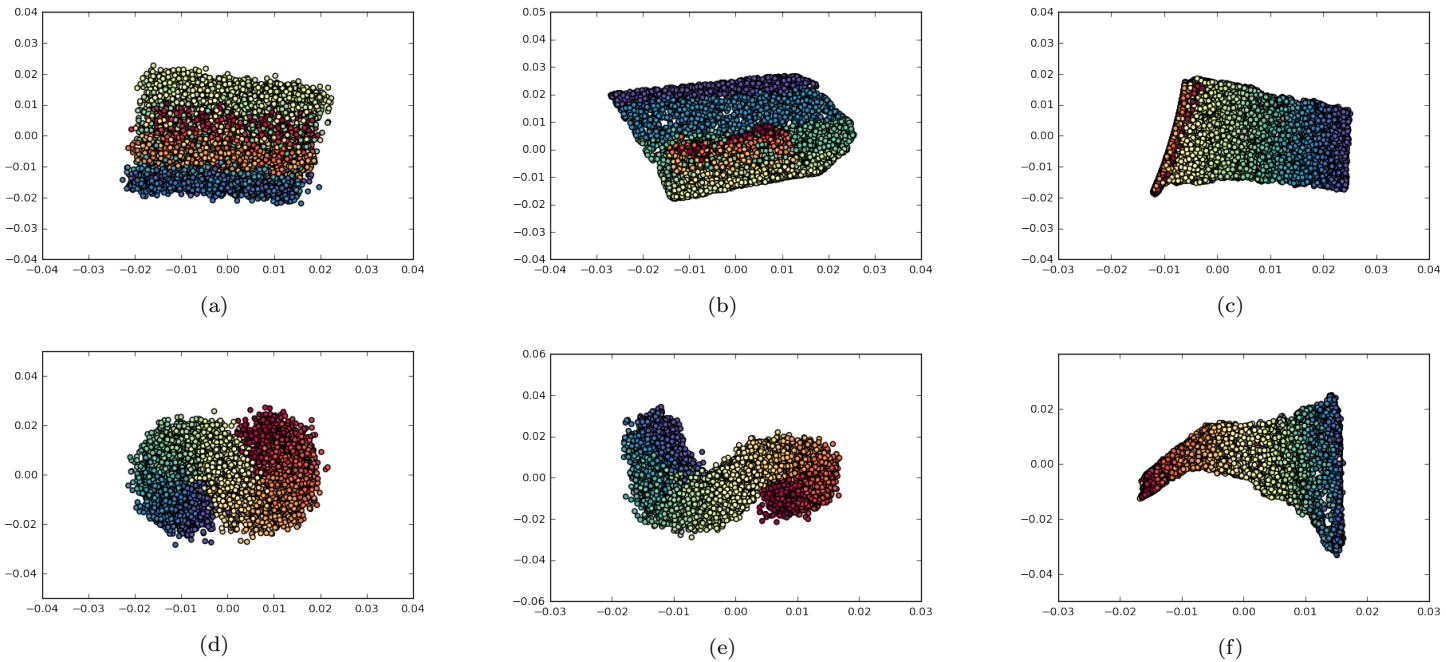


Figure 2.4: Dimension reduction results using the LLE of S-curve and Swiss roll for noised data and projected data through SVDDs and DAEs: (a) noised data of Swiss roll, (b) projected data of Swiss roll using the SVDD, (c) projected data of Swiss roll using the DAE, (d) noised data of S-curve, (e) projected data of S-curve using the SVDD, and (f) projected data of S-curve using the DAE.

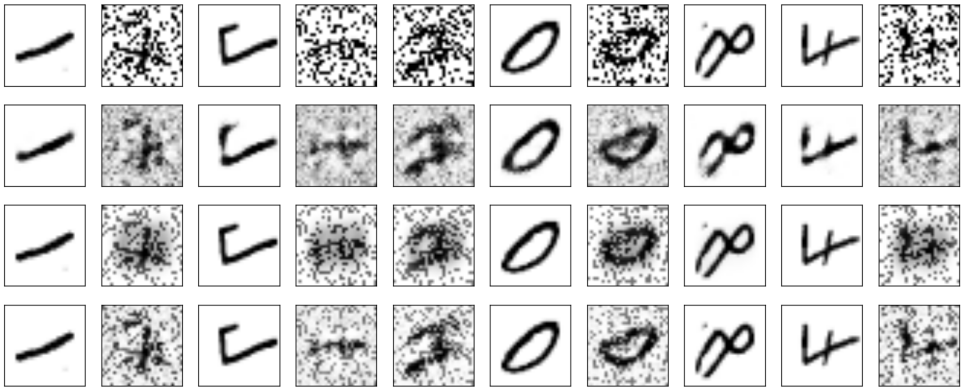


Figure 2.5: MNIST images with 50% noised data. First row presents the original noised images, second row displays the reconstructed images from the trained DAE, third row demonstrates the projected images using the SVDD, and the last row reflects the projected images from the proposed method.

The mapped images using the discrete dynamical system (2.12) smoothed and blurred both the noisy backgrounds and the numeric images. Moreover, the projected images from the SVDD denoised salt-and-pepper noises, but the region around the number became suffused with gray color. The reconstructed images could also destroy the numeric image, as shown in the last column of Figure 2.5, although the reconstructed images from the DAE reduced the salt-and-pepper noises effectively. However, the projected images obtained by our dynamical system (2.10) not only reduced the salt-and-pepper noises but also cleared the backgrounds and numbers in most cases in Figure 2.5. These results imply that our proposed algorithm can approximate a true manifold well and efficiently find more probable points than other methods.

Moreover, we quantitatively evaluated the noise reduction performances in terms of the classification accuracy of the transformed images. In our experiment, we used support vector machine (SVM) classifier, which possessed the rbf kernel with kernel

parameter of 0.01 and the soft margin parameter of 10. We randomly partitioned the test images into two datasets, where one dataset was used for training, and the other dataset was used for testing. We calculated the mean and standard deviation of the classification tasks after repeating these classification tasks 10 times for different partitions and obtaining the accurate results. Figure 2.6 demonstrates the classification results of the MNIST images, where each bar represents the average classification accuracy of a method, and each error bar represents the corresponding standard deviation. The average accuracies, except for the 30% noised data projected by our algorithm, decreased when the ratio of noised data increased. Our proposed method achieved statistically significant best performances on classification tasks for all cases. Specifically, the projected images from the SVDD demonstrated inferior results to the noised data when the ratio of the noised data was high even though the method had the improved performance after reducing the dimensions with the manifold learning methods in K. Kim & Lee (2014a). The results are quite consistent with what we would expect when visualizing examples, where the projection using the SVDD exhibited the blurred numeric images.

We applied our method to the MNIST data by changing the ratio of the noised data. For SVHN dataset, we aim to investigate whether our method is also effective for severe noises without clear images. Therefore, we used 32×32 cropped digit images after converting the RGB images to gray scale and added the masking noises to 10%, 20% and 30% of the pixels of the SVHN gray-scale images to produce a difficult problem. One example in producing the noised images is provided in Figure 2.7, where the first image was a color image, the second image was a gray-scale image, and the degree of noises increased from the third image to the last image. Similar to the experiment of the MNIST images, the test images were projected by our dynamical system and tested by the SVM after we constructed and trained the DAE network with training images. We used the same DAE network and the same

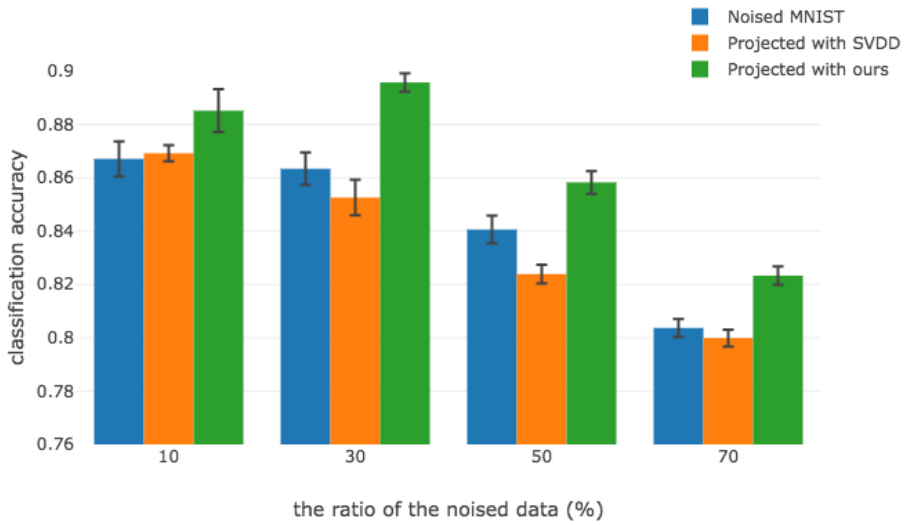


Figure 2.6: Classification accuracies of MNIST data according to the ratio of the noised data.

objective function as the MNIST case, except for the number of input units.

Figures 2.8 depicts the classification performances for the noised SVHN images. The differences in classification accuracies between the noised and projected images increased while the level of noises became severe. In addition, the classification performances of our proposed projection method were significantly superior to SVDD-based nonlinear projection method in all cases. From this fact, we found that our proposed projection method could reduce noises even if only the corrupted images were used instead of the clean images in training phase.

Our proposed projection method implicitly learns the corrupted density distribution smoothed with Gaussian noises, and the approximated manifold is found by following the dynamical system that is defined by the score function of the corrupted density. In addition, in real-world data, the manifold can be captured despite training

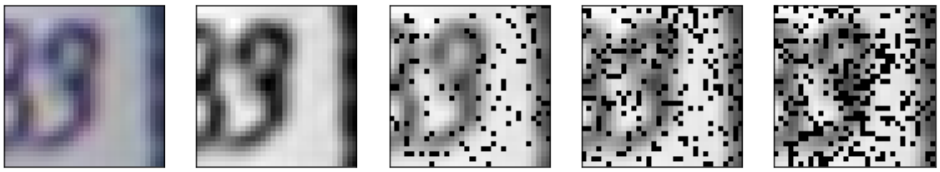


Figure 2.7: Example of the original SVHN image and its variations.

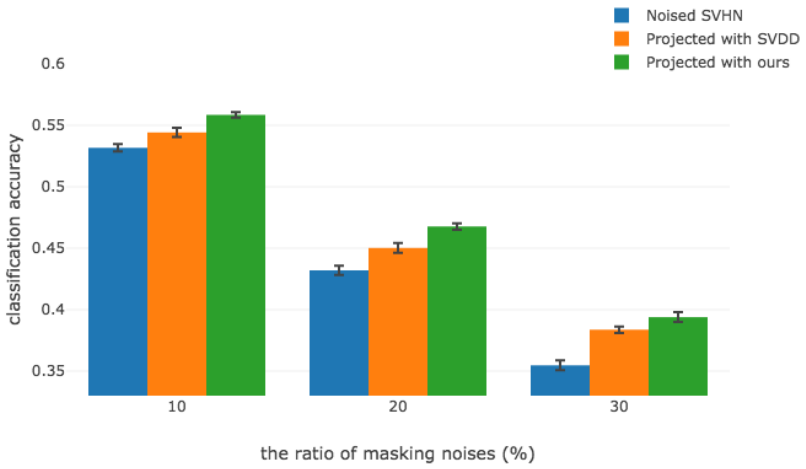


Figure 2.8: Classification accuracies of the SVHN according to the level of masking noises.

the model with corrupted data. The projected images through our proposed method can easily find favorable features and result in enhanced classification performances because the points in the same manifold can be considered as sharing the consistent features.

2.6 Chapter Summary

In summary, this chapter introduced the density-based approach that learns a manifold which is regarded as high-density region. Using a dynamical system, the high-density region (manifold) was identified.

We first developed a stability analysis of the DAE with Gaussian noises when the input density is defined as a *distribution on manifold*. The score function of the corrupted input distribution on the manifold with Gaussian noises indicated the difference between the corrupted input and the reconstruction value of the trained DAE model, which extended the previous analysis results of DAE models with positive input density everywhere. We demonstrated that the high-density region in the corrupted input space corresponds to the stable equilibrium manifold of the DPS associated with a trained DAE model and converges to the true data manifold as the noise level decreases. We also proved that on some mild conditions the attracting equilibrium manifold of DPS is completely stable. From this analytical result, we developed an effective scheme of nonlinear projection using the DAE with Gaussian noises, which was used not for obtaining good features but for learning the input density implicitly. We trained the DAE model in the training phase, and the test data were nonlinearly projected near the data manifold. The experimental results of toy examples and real-world datasets showed that our proposed method was able to find the complex-shaped manifold effectively, to reduce noises by projecting data near the manifold and finally to improve the classification performances.

Chapter 3

Inductive ensemble clustering and low-density regularization with SVDD

3.1 Chapter Overview

Energy-based unsupervised models learn an energy function that associates low energies to training data and high energies to unobserved data. In other words, energy function has low values near data manifolds. DAE model also learns an energy landscape but has trouble in having the high energies of the low-density region between the adjacent manifolds. Meanwhile, SVDD model is proficient in discriminating and separating the support regions of the input data, so it was effectively applied to non-probabilistic tasks such as clustering and outlier detection (Jung et al., 2010; J. Lee & Lee, 2005, 2006; Ben-Hur et al., 2001; Tax & Duin, 1999; K. Kim et al., 2015). This property can be used to regularize the energy for the low-density region.

SVDD model also constructs a kernel support function from input representations to cluster the data and a pseudo input density from this function. However, the

performance of SVDD model highly depends on kernel similarity measure. Given the multiple clustering results, aggregating the results can be helpful to obtain the improved and robust clustering result. Most of ensemble clustering methods cannot obtain a cluster label for a new point because the models do not have the density estimates. If we can construct kernel support function of new similarity measure from multiple clustering results, the induction for a new point can be possible.

In this chapter, an inductive support vector ensemble clustering method is developed constructing the support function from co-association matrix of multiple clustering results in section 3.2. In Section 3.3, we propose new regularization methods of DAE for low-density separation. In Section 3.4, we summarize the results of this chapter.

3.2 Inductive Ensemble Clustering with Kernel Radius Function

Dividing data into non-convex clusters of the same type is very difficult in unsupervised learning and is susceptible to noise inherent in original data. Also since the necessity to protect confidential personal information is increasing, access to raw data becomes impossible, and only basic partitioning results reporting relation between objects can be obtained. To solve these problems, recent ensemble clustering, also called consensus clustering, attracts increasing attention as it combines basic partitions and provides robust clustering results by capturing clusters of more complex shapes (Strehl & Ghosh, 2002; Fred & Jain, 2005; Zhong et al., 2015; Liu et al., 2015, 2016; Ben-Hur et al., 2001). However, most of the existing ensemble clustering methods need to pre-fix the number of anticipated clusters and cannot perform inductive reasoning on out-of-sample data.

In order to deal with these problems, we propose a new inductive ensemble clustering algorithm using basic partitions and dissimilarity between objects instead of

original data. The proposed method utilizes and refines a co-association matrix (rCM) that combines the several basic partitions from the k-means algorithm as in Liu et al. (2016). With kernel support matching, this approach approximates support for data distribution described by the rCM. The mehod then find the representative points of each cluster and cluster out-of-sample data by analyzing the phase characteristics of the constructed support.

3.2.1 Inductive Support Vector Ensemble Clustering

Consensus Ensemble

In this study, we aggregate the clustering results using CM, whose elements represent the number of co-occurrences in basic partitions. Let $\mathcal{X} = \{o_1, o_2, ..., o_n\}$ be the set of n observations (or objects). Suppose that we have p clustering results from the base partitions $P_1, ..., P_p$ where $P_k : \mathcal{X} \mapsto \{1, 2, ..., b_k\}$ is the partition function. The original CM, $C \in \mathbb{R}^{n \times n}$, is defined as

$$C_{i,j} = \frac{1}{p} \sum_{k=1}^{p} \delta(P_k(o_i), P_k(o_j)), \delta(a, b) = \begin{cases} 1, & \text{if} \quad a = b \\ 0, & \text{if} \quad a \neq b \end{cases}.$$

This CM can be regarded as a similarity matrix or a kernel matrix, but it gives the same weight for all pairs in the same cluster. Zhong et al. (2015) differentiated the weights depending on the distances of the pairs. We generalize this weights so as to enable to use any dissimilarity measure instead of the distances of the original data. They specified the weight directly from a dissimilarity measure $DS(o_i, o_j)$:

$$w_{i,j}(k) = \begin{cases} 1 - \frac{DS(o_i, o_j)}{L_{i,j}^k}, & \text{if} \quad P_k(o_i) = P_k(o_j) \\ 0, & \text{if} \quad P_k(o_i) \neq P_k(o_j) \end{cases}, \tag{3.1}$$

where $L_{i,j}^k$ is the maximum dissimilarity of two points in the same cluster with o_i and o_j for the partition k_{true}. From this weight, the refined CM (rCM), $\tilde{C}_{i,j}$, is

$$\tilde{C}_{i,j} = \frac{1}{p} \sum_{k=1}^{p} w_{i,j}(k). \tag{3.2}$$

Inductive Ensemble with Kernel Support Matching

Our work is motivated from the drawbacks of most consensus clustering methods that suffer from choosing the number of clusters and assigning cluster labels for new data points. We propose a new clustering ensemble framework using SVDD in Tax & Duin (1999).

SVDD model can disentangle the support of the observed data $\mathbf{x}_1, ..., \mathbf{x}_n \in \mathbb{R}^d$ from the remaining region by looking for the smallest enclosing sphere of the radius R through nonlinear transformation Φ:

$$\begin{aligned} \min_{R,\mathbf{a},\xi} \quad & R^2 \\ s.t. \quad & \|\Phi(\mathbf{x}_i) - \mathbf{a}\|^2 \leq R^2 + \xi_i \\ & \xi_i \geq 0 \, \forall i = 1, \ldots, n \end{aligned} \tag{3.3}$$

where $\mathbf{a}$ is the center and ξ is the slack variable.

First, a rCM is calculated from the results of basic partitions. Then with a nonlinear transformation Φ from the input space to some high dimensional space, we find the smallest enclosing sphere from the problem (3.3). In order to solve the equation (3.3), we replace its dual problem (3.8) with the following problem by using rCM instead of kernel function:

$$\begin{aligned} & \max_{\beta_j} \ \sum_j \beta_j - \sum_{i,j} \beta_i \beta_j \tilde{C}_{i,j} \\ & \texttt{s.t.} \ 0 \leq \beta_j \leq C, \ \sum_j \beta_j = 1, \ \forall j. \end{aligned} \tag{3.4}$$

where the inner products of $\Phi(o_i) \cdot \Phi(o_j)$ are replaced with a rCM value $\tilde{C}_{i,j}$. By solving the equation (3.4), the following trained kernel support function can be used for estimating support of a data distribution:

$$\begin{aligned} R^2(o_*) &= \|\Phi(o_*) - \mathbf{a}\|^2 \\ &= 1 - 2 \sum_{j \in SV} \beta_j \tilde{C}_{*,j} + \sum_{i,j \in SV} \beta_i \beta_j \tilde{C}_{i,j}, \end{aligned} \tag{3.5}$$

Here, those optimal points corresponding to the optimal objective value $\hat{R}^2$ with $0 < \beta_j < C$ are called support vectors (SVs), those points with $\beta_j = C$ are called bounded support vectors (BSVs), and SV is an index set of SVs and BSVs. The level set $L_{R^2}(\hat{R}^2) = \{o_* : R^2(o_*) \leq \hat{R}^2\}$ divides the data set into a number of connected components from which the number of clusters is naturally determined. It also captures arbitrary clusters by providing a new similarity relationship between given data points aggregated from the primary basic partitions. To assign the cluster labels of training objects o_j, however, we need to evaluate $R^2(o_*)$ at any object o_*. Since the information of $\tilde{C}_{*,j}$ is not available, it is not possible to apply the support-based clustering methods as in Ben-Hur et al. (2001); Jung et al. (2010); K. Kim et al. (2015); Asuncion & Newman (2007) directly to ensemble clustering.

To overcome this problem, we propose a kernel support matching method for inductive ensemble clustering. Our method starts from the metric representations of the data objects, or we can apply any available non-metric multi-dimensional scaling methods or its extensions to transform the data object o_j to its metric representation $x_j \in \mathbb{R}^d$. We next approximate the support function R^2 with the following kernel support function of the form.

$$\tilde{f}(x) = 1 - 2\sum_j \beta'_j k(x_j, x) + \sum_{i,j \in SV} \beta_i \beta_j \tilde{C}_{i,j}, \tag{3.6}$$

Here $k(\cdot, \cdot)$ is a kernel function and we've used in this study the Gaussian radial basis function (rbf) kernel given by $k(x_i, x_j) = k(\|x_i - x_j\|) = e^{-q\|x_i - x_j\|^2}$. To this end, we first fit the rbf kernel q to preserve the kernel similarity with that of rCM for each pair of (x_i, x_j). Specifically, we use the least-squares method to find q that minimizes the replace the Gram matrix $\min_q \|K - \tilde{C}\|$ where $\tilde{C}$ is the Gram matrix of the rCM with $\tilde{C}_{i,j}$ and K is the kernel matrix with $K_{i,j} = k(x_i, x_j)$. The new kernel makes it possible to get the similarity between a given data object and a new data object. With this fitted kernel, we then approximate the kernel support function of (3.6) to

that of (3.5) by calibrating the coefficients of the second term in (3.6) to solve

$$\min_{\beta'} \sum_{i \in SV} \left(\sum_{j \in SV} (\beta_j \tilde{C}_{i,j} - \beta'_j K_{i,j}) \right)^2, \tag{3.7}$$

Finally, we locate the stable equilibrium vectors (SEVs) of the dynamical system associated with this matched kernel support $\tilde{f}$ as in K. Kim et al. (2015). The SEVs in the same connected component of the level set $L_{\tilde{f}}(\hat{R}^2) = \{x : \tilde{f}(x) \leq \hat{R}^2\}$ will be assigned to the same cluster label. When the system is applied to data, each data object converges to one of the SEVs and the same cluster label will be assigned to it.

One of the salient features of this method is inductive clustering. Any unknown new data object belongs to one of the SEV basins and can be assigned to the same cluster label of the corresponding SEV. This labeling process can be expedited by adopting the fast phase as in Asuncion & Newman (2007). With this procedure, the entire data sample space can be divided into several cluster regions, allowing for inductive clustering processing. The procedure of the proposed method (IECS) can be summarized as follows:

1. (co-association) Perform clustering with p basic partitions, and get the rCM using (3.2).

2. (fitting the kernel parameter) Optimize (3.4) with the rCM, and fit a kernel parameter (q in a rbf kernel) via the least squares method.

3. (kernel support matching) Obtain the kernel support (3.6) matching to that of (3.5) by solving (3.7).

4. (cluster labelling) Locate the SEVs of the dynamical system associated with (3.6) and assign the same labels to the SEVs belonging to the same connected component of the level set $L_{\tilde{f}}$.

5. (inductive clustering) Assign to each data object (given training data or unknown test data) the same cluster label of its corresponding SEV.

3.2.2 Experimental Results

To evaluate the performance of the proposed method, we used a number of real-word data sets from the UCI repository (Hubert & Arabie, 1985). The detailed descriptions of datasets are given in table 3.1. K-means algorithm, one of the widely-used clustering algorithms, is used to generate the basic partitions. We set a dissimilarity measure to a distance between data. We compared the proposed method with state-of-the arts ensemble clustering methods such as Cluster-based Similarity Partitioning Algorithm (CSPA), Hyper-Graph Partitioning Algorithm (HGPA), Meta-Clustering Algorithm (MCLA), and spectral clustering of the rCM (SPC) (Strehl & Ghosh, 2002; Zhong et al., 2015; Liu et al., 2016). We obtained 100 set of basic partitions where the number of clusters ranges from the true cluster number k_{true} to $\sqrt{n}$ since the compared methods need to set the numbers of final clusters before the experiments whereas our method does not.

Table 3.1: The descriptions of the datasets

Datasets	Clusters(k_{true})	Instances (n)	Dimensions (d)
Orange	9	140	2
Two circles	2	300	2
Iris (UCI)	3	150	4
Glass (UCI)	6	214	9
Zoo (UCI)	7	101	16
WPBC (UCI)	2	198	32
Satimage (UCI)	6	6435	36

In our experiments, we implemented our algorithm in MATLAB and adopted the adjusted Rand index (ARI) as the cluster evaluation measure (Hubert & Arabie, 1985). The higher the ARI is, the better a quality of clustering is. Table 3.2 reports

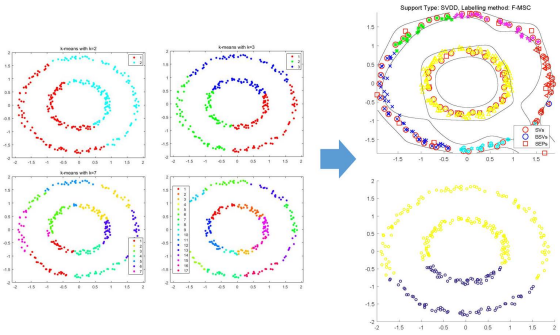


Figure 3.1: K-means and ensemble clustering results for two circles data. The left four graphs are the basic partitions from k-means, the top right is the result of IECS, and the bottom right is the result of SPC.

the clustering performance results of the compared methods. For each data set, the first row reports the ARI of the compared method when the true cluster number is known (the best result is underlined) and the second row when it is not known (the best result is bold-typed). Our proposed method outperformed the other ensemble clustering algorithms in most cases, and it had comparable performances to MCLA with true cluster number is given a priori in Zoo and Smimage datasets. The results show that our method performs very well in real-world problems when the number of true clusters is unknown. Figure 3.1 show a typical example that IECS can detect non-convex shaped clusters well compared with the other ensemble algorithm even though the basic partitions can only address convex shaped data.

In order to verify whether IECS can perform inductive clustering, we used the data set consisting of five Gaussians with 1000 instances. We split the data set into a training set and a test set and changed the ratio between them to compare the results. Table 3.3 shows that IECS works very well despite a small amount of training data

Table 3.2: Performance comparisons in ensemble clustering by ARI

Dataset	SPC ($\sqrt{n}$)	CSPA ($\sqrt{n}$)	HGPA ($\sqrt{n}$)	MCLA ($\sqrt{n}$)	IECS
Orange	0.5605 0.8026	0.5969 0.5607	0.6380 0.6741	0.8549 0.8318	**0.8951**
Two Circles	0 0.0912	0 0.0151	0 0.0167	0.0021 0.0221	**0.6179**
Iris	0.5923 0.3699	0.6004 0.2067	0.6530 0.2620	0.5667 0.4454	**0.6949**
Glass	0.1459 0.1566	0.1726 0.1138	0.1797 0.0937	0.1663 0.1287	**0.2100**
Zoo	0.5603 0.4348	0.4430 0.3489	0.4166 0.3653	0.8135 0.6109	**0.7781**
Smimage	0.5415 0.4140	0.3716 0.3427	0.3841 0.3414	0.5774 0.4595	**0.5656**
WPBC	0.0164 0	0.0136 0.0011	0.0063 0.0038	0.0110 0	**0.0524**

Table 3.3: The inductive performances of IECS changing the test ratio

Test ratio(%)	20	40	60	80	100
ARI	0.8925	0.9120	0.8881	0.8997	0.8533

3.3 Low-density Regularization of Denoising Autoencoder with Kernel Radius Function

3.3.1 Necessity of Low-density Regularization

Energy-based models aim to learn the energy surface of data that has low energy value near the manifolds and high energy value otherwise (Boureau et al., 2007). Energy-based models do not require the normalization constraint, so they have more flexible design of model (LeCun et al., 2006). Given training data, many unsupervised models minimize the energy value of training points while the energy value of unobserved data is pulled up. Therefore, where to pull up is one of the most important issues in energy-based unsupervised learning.

As explained in Section 2, DAE can implicitly learn the density function by estimating the score function (Alain & Bengio, 2014), which is proportional to the gradient of the energy function through Gibbs distribution (Boureau et al., 2007). DAE model learn the energy surface by making the energy function as constant as possible through contracting the score near the observed data. However, although DAE can discover the manifold structure, it has difficulty in capturing low density region between disconnected manifolds because only the score of corrupted inputs can be estimated in case of a positive corrupted level as in equation (2.10). Failure of pulling up the energy in this low-density region can lead to the spurious problem in Alain & Bengio (2014).

Low-density separation problem has motivated many semi-supervised representation learning and clustering algorithms (Ben-David et al., 2009; Chapelle & Zien, 2005; Narayanan et al., 2007; Weston et al., 2012; Rifai, Dauphin, et al., 2011). Semi-supervised models used the label information to find low-density region based on natural clustering assumption that data on each connected manifold have the same label (Goodfellow et al., 2016). Some clustering algorithms divide the input space into

several separate regions by finding low density regions (Narayanan et al., 2007; J. Lee & Lee, 2005, 2006; Jung et al., 2010). However, these models rarely had an interest in learning the input density well because they were mainly focused on prediction or classification tasks.

Recently, energy-based generative models without an explicit density function have been developed to estimate the input density implicitly or to obtain samples having the similar density with input data (Z. Dai et al., 2017; T. Kim & Bengio, 2016; Zhao et al., 2016). These models adversarially learn an energy function (an discriminator) which maps each points to a single scalar value and a generator which produces fake samples similar to real data samples. Zhao et al. (2016) proposed the discriminator loss using a margin loss, but they only obtained the weak support discriminator that could distinguish the support of input data from the outside (Z. Dai et al., 2017). Z. Dai et al. (2017) emphasized that the data density should be estimated within the support, and the discriminator should be penalized outside the support. However, they also could not clarify how the model should work outside the support.

According to D. Lee & Lee (2007), a trained kernel support function from a SVDD model can be used to construct a pseudo density function. Also, this function is trained by looking for the smallest enclosing sphere of training data, and, therefore, can discover the support of input data. J. Lee & Lee (2005) proved that dynamical trajectories of the trained support function approach a stable equilibrium point (SEP) within the support. Therefore, we used the SVDD model to pull up the energy of low-density region because the value of the trained pseudo density function becomes low in this region.

In this study, we develop DAE with low-density regularization using kernel radius function. This model helps not only to capture a manifold structure but also to separate low density-region between manifolds. In the following section, we introduce our regularization term and its effect.

3.3.2 Proposed Method

In SVDD model, the optimal solution of the problem (3.3) is obtained by solving the following dual problem:

$$\begin{aligned} &\max_{\beta_j} \ \sum_j \beta_j - \sum_{i,j} \beta_i \beta_j k(\mathbf{x}_i, \mathbf{x}_j) \\ &\texttt{s.t. } 0 \le \beta_j \le C, \ \sum_j \beta_j = 1, \ \forall j. \end{aligned} \tag{3.8}$$

where C is a margin parameter, and the rbf kernel is usually used as $k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-q\|\mathbf{x}_i - \mathbf{x}_j\|^2\right)$. Under some mild condition on q and C, the following pseudo density function asymptotically converges to the input density $p_{data}(\mathbf{x})$ by Theorem 1 of (D. Lee & Lee, 2007):

$$\hat{p}(\mathbf{x}) = \left(\frac{q}{\pi}\right)^{\frac{d}{2}} \sum_i \beta_i k(\mathbf{x}_i, \mathbf{x}) \tag{3.9}$$

where β_i is nonzero only for support vectors (SVs). Therefore, we obtain the trained energy function from equation (3.9):

$$\mathcal{E}_{supp}(\mathbf{x}) = -\log \hat{p}(\mathbf{x}) = -\log \sum_{i \in SV} \beta_i k(\mathbf{x}_i, \mathbf{x}) \tag{3.10}$$

As in J. Lee & Lee (2006), we can distinguish the support from the outside by defining the level set of the energy function (3.10) as $L_{\mathcal{E}}(\hat{e}) = \{x : \mathcal{E}_{supp}(\mathbf{x}) \le \hat{e}\}$, where $\hat{e}$ can be determined as $\hat{e} = \max\{\mathcal{E}_{supp}(\hat{\mathbf{x}}) : \hat{\mathbf{x}} \in SV\}$. The level set $L_{\mathcal{E}}(\hat{e})$ can approximate the support of the data distribution $supp(p_{data}) = \{\mathbf{x} : p_{data}(\mathbf{x}) > 0\}$. The energy surface from equation (3.10) tends to be spiky and has difficulty in capturing the maximal probability ridge unlike DAE because it has several points around the modes of the Gaussian kernels. We utilize the level set of the model rather than the energy surface within the support.

Also, we can construct the dynamical system using the energy (3.10).

$$\frac{d\mathbf{x}}{dt} = -\bigtriangledown_{\mathbf{x}} \mathcal{E}_{supp}(\mathbf{x}) = \bigtriangledown_{\mathbf{x}} \log \sum_{i \in SV} \beta_i k(\mathbf{x}_i, \mathbf{x}) \tag{3.11}$$

The dynamical system (3.11) induces that all trajectories converge to SEPs within the level set $L_{\mathcal{E}}(\hat{e})$. In addition, the dynamical system (3.11) can accelerate the rate of convergence outside the support while the dynamical system of the kernel support function suffers from slow convergence because of the flat structure outside the support (J. Lee & Lee, 2006).

In section 2, we found that DAE model can implicitly estimate the density function that is defined as a distribution on manifold. However, DAE does not place any constraint on the low-density area between manifolds. Therefore, we propose two regularized methods using the support discriminator from the trained energy (3.10).

First, we learn low-density separate energy by using the support discriminator. The detailed procedure is described in algorithm 2. Given the observed data, samples are drawn to obtain SVDD model, since it is sufficient to capture the disconnected structure with only the samples. We decompose the input space into the basins and cluster them using the level set of the energy (3.10). Based on this results, we construct the support discriminator $\mu(\mathbf{x})$ which determines where a point in the input space belong to. Different DAE models are learned for each region in order to capture the disconnectedness of manifolds and estimate the density on each manifold. For test, the points are allocated to regions by the support discriminator and then follow the DAE model for the regions as in Section 2.3.

Second, we develop the support-regularized DAE model which does not directly separate the region but regularize the reconstruction $r(\mathbf{x})$ with the energy (3.10) and the estimated support $L_{\mathcal{E}}(\hat{e})$. By equation (2.12), after applying the reconstruction function, we expect the energy of the reconstructed input to be lower than before and not to fall into the spurious region. The proposed loss is as follows:

$$\mathcal{L}(r) \;=\; \mathbb{E}\left[\|r(\tilde{\mathbf{x}}) - \mathbf{x}\|^2 + \lambda \max(0, \mathcal{E}_{supp}(r(\tilde{\mathbf{x}})) - \hat{e})\right] \tag{3.12}$$

where λ combines the reconstruction loss and the margin loss from the estimated

Algorithm 2 Support-discriminative DAE algorithm

A1 Training the SVDD
Given the observed data $\mathcal{D} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ on some disconnected manifolds $\mathcal{M}_1, \ldots, \mathcal{M}_C$.
Draw samples $\mathbf{z}_1, \ldots, \mathbf{z}_s$ from the observed data $\mathcal{D}$.
Train the SVDD model with samples by optimizing (3.8).
Construct the energy function (3.10), decompose the input space into C regions and obtain the support discriminator $\mu(\mathbf{x}) : \mathbb{R}^d \to \{1, 2, \ldots, C\}$.

A2 Training DAEs for separate regions
Divide the training data $\mathcal{D}$ into $\mathcal{D}_1, \ldots, \mathcal{D}_C$ sets
s.t. $\mathcal{D}_j = \{\mathbf{x}_i | \mu(\mathbf{x}_i) = j, \mathbf{x}_i \in \mathcal{D}\}$
Train the DAE A_j and the reconstruction function $r_j(\mathbf{x})$ with $\mathcal{D}_j$

A3 Testing for new point
Given a new point $\tilde{\mathbf{x}}$, apply the model $A_{\mu(\tilde{\mathbf{x}})}$.

energy $\mathcal{E}_{supp}$. The loss (3.12) means that the model follows the original DAE objective for the support region $supp(p_{data})$ while the model minimizes both the reconstruction error and the estimated energy $\mathcal{E}_{supp}$ for the remaining regions. We estimate the support region with the level set $L_{\mathcal{E}}(\hat{e})$. The loss $\mathcal{L}(r)$ is minimized for $r(\tilde{\mathbf{x}}) = \mathbb{E}^{\mathcal{M}}_{x|\tilde{\mathbf{x}}}$ as in equation (2.8) within the support region, that is, $r(\tilde{\mathbf{x}}) \in L_{\mathcal{E}}(\hat{e})$. In case of $r(\tilde{\mathbf{x}}) \notin L_{\mathcal{E}}(\hat{e})$, the loss $\mathcal{L}(r)$ can be represented as:

$$\begin{aligned}
\mathcal{L}(r) &= \mathbb{E}\left[\|r(\tilde{\mathbf{x}} - \mathbf{x})\|^2 + \lambda \mathcal{E}_{supp}(r(\tilde{\mathbf{x}}))\right] \\
&= \mathbb{E}[\|r(\tilde{\mathbf{x}}) - \tilde{\mathbf{x}} + \tilde{\mathbf{x}} - \mathbf{x}\|^2] + \lambda \mathbb{E}[\mathcal{E}_{supp}(r(\tilde{\mathbf{x}}))] \\
&= \mathbb{E}[\|r(\tilde{\mathbf{x}}) - \tilde{\mathbf{x}}\|^2] + 2\mathbb{E}[(r(\tilde{\mathbf{x}}) - \tilde{\mathbf{x}})(\tilde{\mathbf{x}} - \mathbf{x})] \\
&\quad + \mathbb{E}[\|\tilde{\mathbf{x}} - \mathbf{x}\|^2] + \lambda \mathbb{E}[\mathcal{E}_{supp}(r(\tilde{\mathbf{x}})] \\
&= \mathbb{E}[\|r(\tilde{\mathbf{x}}) - \tilde{\mathbf{x}}\|^2] + \mathbb{E}[\|\tilde{\mathbf{x}} - \mathbf{x}\|^2] + \lambda \mathbb{E}[\mathcal{E}_{supp}(r(\tilde{\mathbf{x}}))] \qquad (3.13)
\end{aligned}$$

When the dynamical system (2.10) is constructed from the optimum $r^*(\mathbf{x})$ of equation (3.12), a trajectory outside the support can be accelerated to get into the support as equation (3.11) and a trajectory within the support can proceed to maximal density

region (manifold). Therefore, our support-regularized DAE model can pull up the energy outside the support and identify multiple manifolds. In the following section, these facts are validated by the illustrative examples.

3.3.3 Illustrative Experiments

In this section, we evaluate our proposed methods through two-dimensional illustrative examples which have the maximal probability ridge and the observed data dispersed around the ridge. The examples are presented in Figure 3.2 (a) and (d). Both of our methods require to train the SVDD model from the problem (3.8), and from the result the energy (3.10) is constructed. The trained energy surfaces are shown in Figure 3.2 (b) and (e) where the support boundary is denoted by the solid red line, and the SEPs of the dynamical system (3.11) are denoted by the red "x".

Support-discriminative DAE model first decomposes the input space into several regions, following Algorithm 2. We used the cluster labeling method in Jung et al. (2010) for the level set $L_{\mathcal{E}}(\hat{e})$, and the clustering results are presented in Figure 3.2 (c) and (f). From this figure, we can indirectly find the discriminative boundary of $\mu(\mathbf{x})$ in Algorithm 2. Support-regularized DAE model utilizes the support boundary in Figure 3.2 (b) and (e) to construct the loss (3.13). Both of the illustrative examples have the maximal probability ridge, but the trained energy (3.10) from the SVDD model cannot capture this ridge but have the spiky surfaces within the support boundary (see Figure 3.2 (b) and (e)). We addressed this problem by using DAE model because the DAE model can capture a maximal probability ridge well as shown in Section 2.5. To compare the energy, we visualize the learned vector fields $r(\mathbf{x}) - \mathbf{x}$ and the reconstructed inputs from the noised data in Figure 3.2 (c) and (f), where applying the reconstruction function $r(\mathbf{x})$ has an effect of using the discrete dynamical system (2.12). Figure 3.3 shows the reconstruction results from DAE, support-discriminative DAE and support-regularized DAE models.

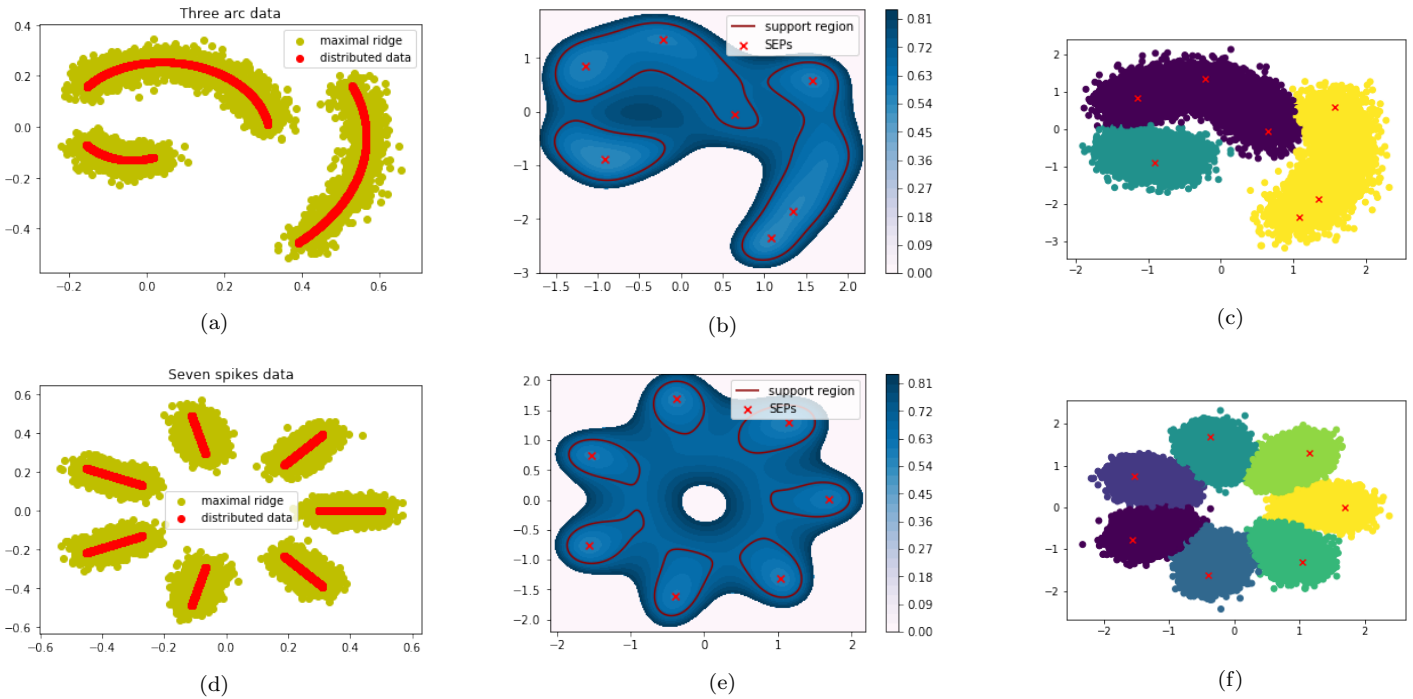


Figure 3.2: Two illustrative examples: (a) three arcs data, (b) the energy contour of three arcs data, (c) clustering result for the noised data of three arcs data, (d) seven spikes data, (e) the energy contour of seven spikes data, and (f) the clustering result for seven spikes data.

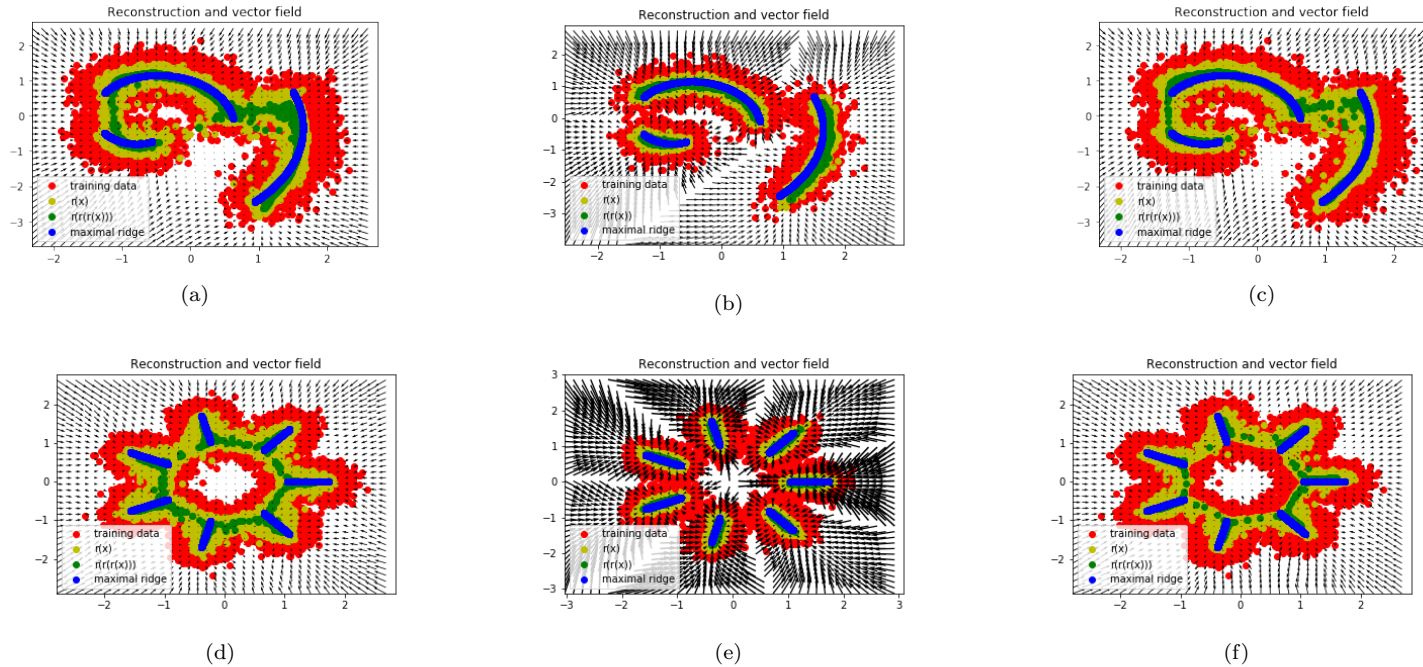


Figure 3.3: Reconstruction results and learned vector fields of models, where red points are the noised data, yellow and green points are the reconstruction results, and the blue points lie on the true maximal probability ridges: (a) and (d) are the results of DAE model, (b) and (e) are the results of support-discriminative DAE model, and (c) and (f) are the results of the support-regularized DAE model.

The original DAE model well learned the maximal probability ridge, but it was difficult to raise the energy of the low-density region between two manifolds (see Figure 3.3 (a) and (d)). Meanwhile, support-discriminative DAE model was able not only to capture the maximal probability ridges but also to strongly diverge around the low-density regions (see Figure 3.3 (b) and (e)). Figure 3.3 (c) and (f) shows the results of support-regularized model. We can see that support-regularized model learned the increased energy for the low-density region because the less points lies on the low-density region after applying the reconstruction function than Figure 3.2 (a) and (d).

While the support-discriminative model showed the best performances on finding the maximal probability ridges and pulling up the energy of the low-density region, the performances of the model depend on the clustering results. However, the SVDD model has trouble in clustering a high-dimensional data but an ability to capture the support of the input space. Support-regularized model can be effective even if the SVDD model can capture only the support region. Therefore, we can determine the appropriate regularization method for DAE model to pull up the energy of the low-density region.

3.4 Chapter Summary

In summary, we learned the input density using DAE and SVDD models and constructed the energy in perspective of Gibbs distribution. SVDD model also can be applied to ensemble clustering by the estimated kernel support function.

We presented a new inductive clustering ensemble algorithm by learning the kernel support function. The method aggregates basic clustering results using kernel support matching and automatically decides the effective number of clusters. The trained kernel support function can estimate the density of a new point and make it possible

to inductively allocate a cluster label using a dynamical system. Experimental results showed that the proposed method not only effectively captured non-convex clusters but also made inductive clustering very well for out-of-sample data.

Finally, we regularized the DAE model to improve the low-density separation between the disconnected manifolds. We found the support of the input data using the energy of the SVDD model and proposed two regularization methods. Support-discriminative model decomposes the input space into the basins and combines them as a cluster region if they contain a connected manifold. A support discriminator is constructed to determine cluster regions of points in the input space, and DAE models are trained for each region which contained a connected manifold. Support-regularized model combines a margin loss from the energy of SVDD model with the reconstruction loss of original DAE. The objective of this model behaves similarly to the DAE model defined as a *distribution on manifold* within the support region and it is affected by the negative gradient of the energy of SVDD model outside the support. Because SVDD model could well capture the support region, the separation of the low-density region between the disconnected manifolds was improved in the illustrative examples.

Chapter 4

Semi-supervised Distributed Representation for Sentiment Analysis

4.1 Chapter Overview

Sentiment analysis, which analyzes opinions, sentiments, and attitudes of writers in written documents, is an important issue in natural language processing (NLP) and has been applied to diverse fields such as business analytics, marketing, and finance (Kloptchenko et al., 2004; Gupta & Lehal, 2009; Tated & Ghonge, 2015; Rabhi et al., 2009). Machine learning approaches such as SVM and neural networks have been applied successfully to sentiment classification (Pang et al., 2002; Pang & Lee, 2008; L.-S. Chen et al., 2011). However, documents should be represented as fixed-length vectors to apply machine learning techniques for sentiment analysis.

Distributed representation can induce various similarity relationships without specifying a role of each feature. Distributed representation models of words and documents such as Word2vec and Doc2vec learn continuous numerical vectors from textual

inputs that should be represented as fixed-length vectors to apply machine learning techniques for text data analysis. This continuous representation makes it possible to identify the input density of words and documents. Distributed representation models for words and documents have shown superior performances to dictionary-based representation model which lose the order information of words (Mikolov, Sutskever, et al., 2013; Mikolov, Chen, et al., 2013; Le & Mikolov, 2014). They have proved that distributed representation models are good at capturing the meaningful relationships between documents or words well and free from sparsity and high dimensionality.

However, while distributed representations can capture semantic relationships well, representations cannot distinguish between synonyms and antonyms. In terms of sentiment analysis, this can be a crucial problem because most sentiment words have a synonym or antonym relationship. It can be detrimental because sentiment analysis is an important issue in NLP. Although improved word representations distinguishing synonyms and antonyms were developed, they depended on additional knowledge or information, such as tags of words and lexical contrast information (Scheible et al., 2013; Nguyen et al., 2016; Park, 2016). In addition, for sentiment analysis, many studies have applied neural probabilistic language models to sentiment analysis reflecting the sentiment information to document representations, but these studies needed to parse sentences before learning (Socher et al., 2011, 2013) or added sentiment classifiers to the top of networks as supervised learning (Socher et al., 2011, 2013; Ranzato & Szummer, 2008).

In this section, we propose a semi-supervised distributed representation method that reflects the difference of document distributions depending on the sentiments using partially labeled documents. This is a reasonable approach in the real world because most real-world documents have no sentiment labels and the cost of the labeling process is rather high. Hence, to reflect partially available sentiment information as well as to retain the contexts of word sequences, the proposed method additionally

considers terms related with sentiment labels to the objectives in Le & Mikolov (2014) as well. The proposed method modifies only local structures of document embeddings by considering the sentiment information of neighbor documents for effective and efficient learning. The obtained document representations are expected to become closer if they have similar semantic structures and the same sentiments.

4.2 Distributed Representations

In distributed representation, semantically similar inputs are close to each other in distance. The term "semantically similar" can encompass many different concepts depending on the type of an input. In case of NLP, similar inputs have semantic or syntactic information.

Distributed representations of words or phrases were developed using neural probabilistic models. Neural probabilistic models are able to reflect similarity between words and phrases which typically means that two words are similar if they are used in similar context or occur nearby, and phrases are similar if they share similar words (Mikolov, Sutskever, et al., 2013; Mikolov, Chen, et al., 2013; Le & Mikolov, 2014). From the perspective of distributed representation, the trained vectors have semantic and syntactic relationships. These representations also have additive compositionality, an attractive property that causes the vectors to have a linear relationship with each other and represent the distribution of the context in which a word appears.

In distributed representation learning, every word and document is mapped to a unique vector, which is learned from scratch and not from other numeric vectors. A group of distributed representation learning models of words called Word2vec consists of the skip-gram and Continuous Bag of Words (CBOW) models (Mikolov, Chen, et al., 2013). In the skip-gram model, word representations are trained to predict the surrounding words in a sentence or a document. Meanwhile, the CBOW model learns

word representations by maximizing log probability of a word given the surrounding words. Given a sequence of training words $w_1, w_2, w_3, ..., w_T$, the objective of the skip-gram model is

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t) \tag{4.1}$$

where c is the length of context. The conditional probability of the j-th word given the i-th word $p(w_j|w_i)$ is calculated using the softmax function for the input I and output O as follows.

$$p_\theta(O|I) = \frac{\exp\left(s_\theta(O, I)\right)}{\sum_{j\in\mathcal{O}}\exp\left(s_\theta(j, I)\right)} \tag{4.2}$$

where v_i is an input embedding of the instance i, $\tilde{v}_j$ is an output embedding of the instance j, $s_\theta(j, i) = \tilde{v}_j^\top v_i + b_j$ is a score, and $\mathcal{O}$ is the set of all possible outputs.

The objective of CBOW model is to maximize the average of the log probability

$$\frac{1}{T}\sum_{t=c}^{T-c}\log p(w_t|w_{t-c}, ..., w_{t+c}) \tag{4.3}$$

In equation (4.3), the log probability of a word given the surrounding words is also defined using the softmax function, where the input vector is the concatenation, sum, or average of surrounding word vectors. However, the equation (4.2) is impractical to optimize because of calculating normalization constant for all possible outputs. Distributed representation models significantly reduced this computational cost by sub-sampling of frequent words (Mikolov, Sutskever, et al., 2013) and using the noise contrastive estimation (NCE) (Mnih & Kavukcuoglu, 2013; Gutmann & Hyvärinen, 2010). In the result, these two methods can map words with similar meanings to similar positions in vector space.

Inspired by learning methods for the distributed representations of words, a model learning distributed representations of documents, Doc2vec, was developed by Le & Mikolov (2014). Two approaches to learning a document vector can be used, similar

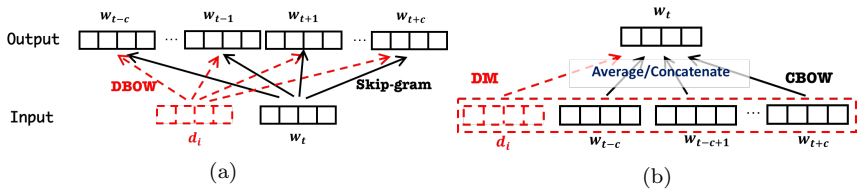


Figure 4.1: The model architectures of distributed models: skip-gram and CBOW models learn word embeddings, and DBOW and DM models learn document embeddings.

to learning a word vector. Figure 4.1 illustrates the model architectures of these distributed models.

First, distributed memory model of paragraph vectors (DM) is affected by CBOW model as in Figure 4.1 (b). In this model, the paragraph vector is added to the conditioning part of the log probability in equation (4.3) with vectors of other surrounding words. The document and word vectors can be learned simultaneously by a similar objective. This model has an advantage because it can consider the word order in the context to obtain document representations unlike the dictionary-based models.

Distributed bag of words (DBOW) model is another distributed representation learning method for documents. This model predicts words in the paragraph, given the paragraph vector as the input and is the opposite of DM model, but is similar to the skip-gram model (Mikolov, Sutskever, et al., 2013) as in Figure 4.1 (b). Moreover, in DBOW model, the word vectors are usually trained by the skip-gram model, where the context information can be considered indirectly. In sentiment analysis, the DBOW model can be useful because the occurrence of sentiment words in a document is important in discriminating sentiment of documents.

These models outperformed count-based models in sentiment analysis and information retrieval tasks as well as semantic texture similarity (STS) task (Le & Mikolov, 2014; Lau & Baldwin, 2016). Distributed representations, however, tend to have dif-

ficulty in reflecting sentiment information. For example, words “good” and “bad” have opposite meanings, but their vector representations are close to each other in a representation space because they tend to occur in a similar context. Therefore, distributed representation can be improved by reflecting the sentiment information, and we address this problem in Section 4.3.

4.3 Proposed Method

As mentioned in Section 4.2, the existing distributed representation learning methods for words tend to have a deficiency in implying sentiment information. This fact may also affect the learning of the distributed representations of documents, making it difficult for document vectors to reflect sentiment information. To solve this problem, we developed an improved method to learn the sentiment embeddings of documents. We assume that partial sentiment labels of documents are given, which is quite reasonable because a labeled training set is necessary to build a sentiment classification model. Using partially labeled documents, we might obtain improved document representations in the sense that documents with similar sentiments will have similar vectors.

In this study, the improved part is based on two assumptions. The first one is a manifold assumption that the learned representations are located on the lower dimensional manifold than the original space. The other assumption is a smoothness assumption of the sentiment classifier on learned representations, which means that documents with similar input representations would have similar conditional probabilities of sentiment labels given the inputs. Therefore, we want to adjust the vector representations of documents depending on their sentiment distribution based on the manifold. The proposed method is based on the DBOW model because the DBOW model considers each word rather than the context and conceives the context infor-

mation through the trained word vectors implicitly, which is more appropriate for sentiment analysis where word information is important.

First, we assume that we have N document corpus $\{d_1, ..., d_N\}$ which consists of N_L labeled documents $\{d_1, ..., d_{N_L}\}$ and N_U unlabeled documents $\{d_{N_L+1}, ..., d_{N_L+N_U} = d_N\}$. From the given documents, we considered the relationship of documents based on local structures and the sentimental information. Because high dimensional data would concentrate on low dimensional manifold, we reflect the sentiment information only from neighboring documents. Using the smoothness assumption, we want to make the documents with the same sentiments congregate. As shown in equation (4.2), we can calculate the conditional probability $P(d_i|d_j)$ between two documents d_i and d_j, and we want to learn document embeddings that have high conditional probability $P(d_i|d_j)$ if d_i and d_j have the same sentiment. We set a label y_j of the document d_j as $+1$ for a positive document, -1 for a negative document, and 0 for an unlabeled. However, because the intractability problem of the denominator still exists, we employed Noise Contrastive Estimation (NCE), which was developed by Gutmann & Hyvärinen (2010) and used for negative sampling of skip-gram model (Mikolov, Sutskever, et al., 2013). Unlike negative sampling, we trained the model to discriminate among documents with respect to the sentiments.

Let $P(S = +|d_i, d_j)$ denote that the documents d_i and d_j have the same sentiment. Conversely, the probability that documents d and d' have the opposite sentiment is $P(S = -|d_i, d_j) = 1 - P(S = +|d_i, d_j)$. We assume that the probability can be parameterized by θ: $P(S = +|d_i, d_j; \theta)$, where θ is the document embeddings. We can define this probability using the sigmoid function:

$$p(S = +|d_i, d_j; \theta) = \sigma({v_{d_i}}^T v_{d_j}) = \frac{1}{1 + \exp\left(-{v_{d_i}}^T v_{d_j}\right)}, \tag{4.4}$$

where we used the original sigmoid function because we assumed that the ratio of positive documents is 0.5. This definition makes sense because the closer the docu-

ments are, the higher the probability of having the same sentiment is. We set the objective to find the document embeddings by maximizing the probabilities that observations having the same sentiment also have the same sentiment and observations having the opposite sentiments have opposite sentiments in the embedding space. The optimization objective of sentiment discrimination was set as follows:

$$
\begin{aligned}
J &= \arg\max_{\theta} \prod_{(d_i,d_j)\in S_+} P(S=+|d_i,d_j;\theta) \prod_{(d_i,d_j)\in S_-} P(S=-|d_i,d_j;\theta) \\
&= \arg\max_{\theta} \sum_{(d_i,d_j)\in S_+} \log\sigma({v_{d_i}}^T v_{d_j}) + \sum_{(d_i,d_j)\in S_-} \log\left(1-\sigma({v_d}^T v_{d'})\right) \\
&= \arg\max_{\theta} \sum_{(d_i,d_j)\in S_+} \log\sigma({v_{d_i}}^T v_{d_j}) + \sum_{(d_i,d_j)\in S_-} \log\sigma(-{v_{d_i}}^T v_{d_j})
\end{aligned}
$$

where S_+ is the set of all pairs with the same sentiment and S_- is the set of all pairs with opposite sentiments.

However, the computational complexity of the above objective would be too high because of the consideration of all pairs in the labeled documents. Moreover, we need to preserve the document information as a sequence of words. Therefore, in training a document embedding, the sentiment objective was computed for only pairs of documents which belong to each other's k-nearest neighbor set and added to the original DBOW objective. It can make the training more efficient and preserve the structure of embedding space. To update the embedding v_{d_i} of document d_i, we used the sub-objective:

$$
\arg\max_{v_{d_i}} \sum_{i\in N(d_j)\ s.t. y_i y_j=1} \log\sigma({v_{d_i}}^T v_{d_j}) + \sum_{i\in N(d_j)\ s.t. y_i y_j=-1} \log\sigma(-{v_{d_i}}^T v_{d_j}), \quad (4.5)
$$

where $N(d_j)$ is the set of neighbors of d_j. Therefore, we also adopted stochastic gradient method to train the word and document vectors, which is generally used for training neural network models.

We developed the semi-supervised DBOW (semi-DBOW) model by adding the sentiment objective term (4.5) to the objective function of DBOW. Because the documents share word embeddings, the unlabeled documents could be affected indirectly by labeled documents. Therefore, using semi-DBOW, we may obtain document embedding that not only reflects the partial sentiment information but also accommodates the effectiveness of DBOW model.

We compared some unlabeled document representations from DBOW and semi-DBOW to determine whether adding the term (4.5) can make the embeddings of unlabeled documents reflect sentiment information. First, we randomly selected five unlabeled documents from Electronics dataset and examined their 10-nearest neighborhood documents, their true labels (y_{true}), and their train labels (y_{train}) from which we can determine whether the document was used as unlabeled or labeled document when training the representations. We trained the document vectors based on the experimental design in Section 4.4.2. If a document has positive sentiment, then its true label is $+1$; otherwise, it is -1. Additionally, the train label of a document has the same value as the true label if the document is used as a labeled document, otherwise, it has a value of 0. Table 4.1 shows five unlabeled documents and their neighbors determined by the representations from DBOW and semi-DBOW model in descending order of distance. The selected five documents consisted of three negative documents and two positive documents. In Table 4.1, the common neighbors between the DBOW and semi-DBOW models are highlighted. From Table 4.1, we found that semi-DBOW can conserve the local structure of original DBOW as expected. For example, DOC2, DOC7, and DOC8 highlighted their neighboring documents over five, which implies that the change in neighbors when semi-DBOW was used instead of DBOW is not large. However, documents with the same sentiment are usually in the top of neighboring documents. This observation showed that document representations from semi-DBOW congregated with documents having the same sentiment,

DOC2[unlabeled/negative]					
DBOW			semi-DBOW		
ID	y_{train}	y_{true}	ID	y_{train}	y_{true}
3908	0	-1	3908	0	-1
1476	0	-1	1476	0	-1
1728	0	-1	1728	0	-1
6438	-1	-1	364	0	-1
7852	+1	+1	4014	0	-1
2473	+1	+1	6283	-1	-1
203	-1	-1	6438	-1	-1
792	+1	+1	62	0	+1
7348	0	+1	7852	+1	+1
4220	0	+1	2281	+1	+1

DOC5[unlabeled/negative]					
DBOW			semi-DBOW		
ID	y_{train}	y_{true}	ID	y_{train}	y_{true}
280	0	-1	280	0	-1
8055	+1	+1	6172	-1	-1
4455	0	-1	406	-1	-1
4277	-1	-1	3903	-1	-1
3625	-1	-1	4001	-1	-1
5638	-1	-1	6711	0	-1
2415	+1	+1	8034	-1	-1
2873	+1	+1	404	0	-1
3334	-1	-1	2403	-1	-1
442	-1	-1	4036	-1	-1

DOC7[unlabeled/positive]					
DBOW			semi-DBOW		
ID	y_{train}	y_{true}	ID	y_{train}	y_{true}
6718	+1	+1	6718	+1	+1
5818	0	+1	4744	+1	+1
4744	+1	+1	2805	+1	+1
7978	+1	+1	7978	+1	+1
2805	+1	+1	5818	0	+1
1521	0	+1	3009	+1	+1
7051	0	+1	1920	+1	+1
5219	+1	+1	67	+1	+1
1920	+1	+1	6693	+1	+1
7368	0	+1	7463	+1	+1

DOC8[unlabeled/positive]					
DBOW			semi-DBOW		
ID	y_{train}	y_{true}	ID	y_{train}	y_{true}
7530	+1	+1	7530	+1	+1
7200	0	+1	84	+1	+1
84	+1	+1	7200	0	+1
1401	0	+1	1411	+1	+1
1411	+1	+1	5108	+1	+1
1661	+1	+1	1547	+1	+1
2224	0	+1	1401	0	+1
1846	0	-1	1661	+1	+1
2814	+1	+1	1707	+1	+1
4483	0	-1	4307	+1	+1

DOC15[unlabeled/negative]					
DBOW			semi-DBOW		
ID	y_{train}	y_{true}	ID	y_{train}	y_{true}
240	0	+1	7034	-1	-1
416	+1	+1	7804	-1	-1
1917	-1	-1	5072	-1	-1
4102	-1	-1	3081	-1	-1
5539	0	-1	1917	-1	-1
5623	+1	+1	7037	-1	-1
210	+1	+1	7413	-1	-1
5470	0	+1	240	0	+1
1527	-1	-1	1664	-1	-1
7955	-1	-1	2138	-1	-1

Table 4.1: Unlabeled documents and their 10-nearest neighbors for DBOW model and semi-DBOW model

although we selected the unlabeled documents. This result might be because most neighbors in semi-DBOW had the same sentiment labels while neighbors in DBOW did not. In addition, the effects of preserving local structure and reflecting sentiment label complemented each other. For example, in the case of DOC2, while positive documents such as 7852 and 2473 were near neighbors of DOC2 in DBOW, they were pushed back on the neighborhood list, but document 7852 was still one of the 10-nearest neighbors. Through this result, we can roughly validate the effects of semi-DBOW objective on conserving neighbors from DBOW and adjusting representations with part of the sentiment labels. In the following section, we conducted experiments of sentiment visualization and classification to approve the effectiveness of our representation in sentiment analysis.

4.4 Experimental Results

4.4.1 Data description

For the experiments, we used two real-world datasets, namely, Amazon review dataset in Blitzer et al. (2007) and Yelp review dataset from *Yelp dataset Challenge*[1]. Amazon

[1] `https://www.yelp.com/dataset_challenge/`

dataset provides product reviews of Amazon.com and Yelp dataset consists of the restaurant reviews of the Yelp site. Among the product reviews of Amazon dataset, we selected the reviews for four categories of products: Book, DVD, Electronics, and Kitchen. In the case of Yelp reviews, we categorized reviews according to years and chose three years: 2008, 2010, and 2013. The datasets used are summarized in table 4.2.

Datasets		Number of documents	Number of sentences	Dictionary size[2]
Amazon	Book	43,142	497,864	43,152
	DVD	33,160	406,000	38,205
	Electronics	8,096	66,987	8,792
	Kitchen	6,238	46,059	7,600
Yelp	Yelp-2008	27,964	213,194	21,255
	Yelp-2010	88,220	632,424	35,082
	Yelp-2013	266,636	1,672,809	52,076

Table 4.2: The summary of datasets

Each review has a score between 1 and 5, where we regarded 1-2 points as negative sentiments and 4-5 points as positive sentiments. We balanced the positive and negative documents because of the assumption that the ratio of positive documents is 0.5 to avoid additional parameter in equation (4.4).

4.4.2 Experimental procedure

We compared our method with two frequency-based methods (BOW and TF-IDF), two latent representation methods (latent semantic index (LSI) and latent Dirichlet allocation (LDA)), two distributed representation methods (DM and DBOW), and one semi-supervised representation method (semi-supervised Laplacian eigenmap (SSLE) (K. Kim & Lee, 2014b)). Because most semi-supervised or supervised representation

[2]Dictionary size is the number of unique words in each corpus without stemming

learning methods are started not from scratch but from unsophisticated numeric vectors, they mapped original document vectors to other transformed vectors with all or part of sentiment labels (K. Kim & Lee, 2014b; Ranzato & Szummer, 2008; Ramage et al., 2009; Mcauliffe & Blei, 2008; Andrzejewski & Zhu, 2009). We selected SSLE as one of comparison methods among the semi-supervised representation methods because it achieved superior performances to other semi-supervised methods in sentiment tasks. We used *Gensim* library[3] for the comparison methods except SSLE. Because the representations of BOW and TF-IDF have very high dimensions, we selected the top words based on occurrence frequency of words to match the dimensions of representations obtained by other methods. LSI representations were learned from TF-IDF, and LDA representations were learned from BOW. For DM, DBOW, and semi-DBOW models, we set the size of embedding as 200, window size as 3, minimum count as 3, and others as default options in *Gensim* doc2vec. We trained the embedding of documents by sentence units. The semi-DBOW model requires more hyper-parameters such as number of neighbors, k, and the learning rate β for equation (4.5). We conducted experiments on Electronics dataset to verify the effect of hyper-parameters and determined the appropriate values based on the results. For SSLE, we followed the same procedure to obtain two dimensional representations of SSLE.

The procedure of the experiment in our model is shown in Figure 4.2. First, we train or calculate the vector representations of documents using all documents and their sentiment labels if required. Next, we evaluate the obtained representations only with vectors of unlabeled ones, D2 in Figure 4.2. We then split these representations into training and test datasets repeatedly and average the test performances. We reduced dimensions of document representations with PCA when two-dimensional vectors are required, such as visualization or two-dimensional sentiment classification.

[3] `https://radimrehurek.com/gensim/`

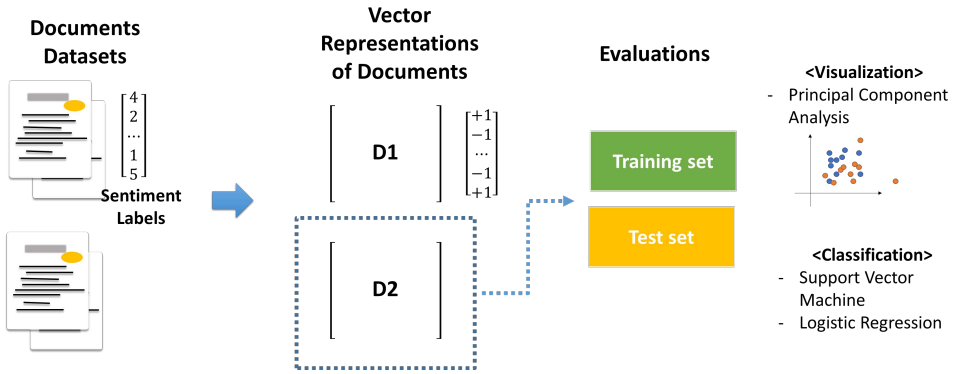


Figure 4.2: Experimental Procedure.

For sentiment classification, we used Logistic Regression (LR) and Support Vector Machine (SVM) as classifiers. In our experiment, we set the rbf kernel with parameter 0.01 and set the soft margin parameter to 10 for SVM. We selected the classification accuracy as a performance measure, and trained and tested the models only with unlabeled documents. We repeated the same procedures 10 times for randomly split datasets. Because SSLE has a memory problem when dealing with large datasets [4] such as Book and DVD datasets, we sampled the documents and trained the model. Nevertheless, as SSLE exhibited bad performance on large datasets, we skipped SSLE in Yelp datasets.

To evaluate performance for both tasks, we used only embeddings from unlabeled documents because other models did not use sentiment information of documents. Therefore, we can determine whether the sentiment information of labeled documents can be helpful to obtain better representations of unlabeled documents.

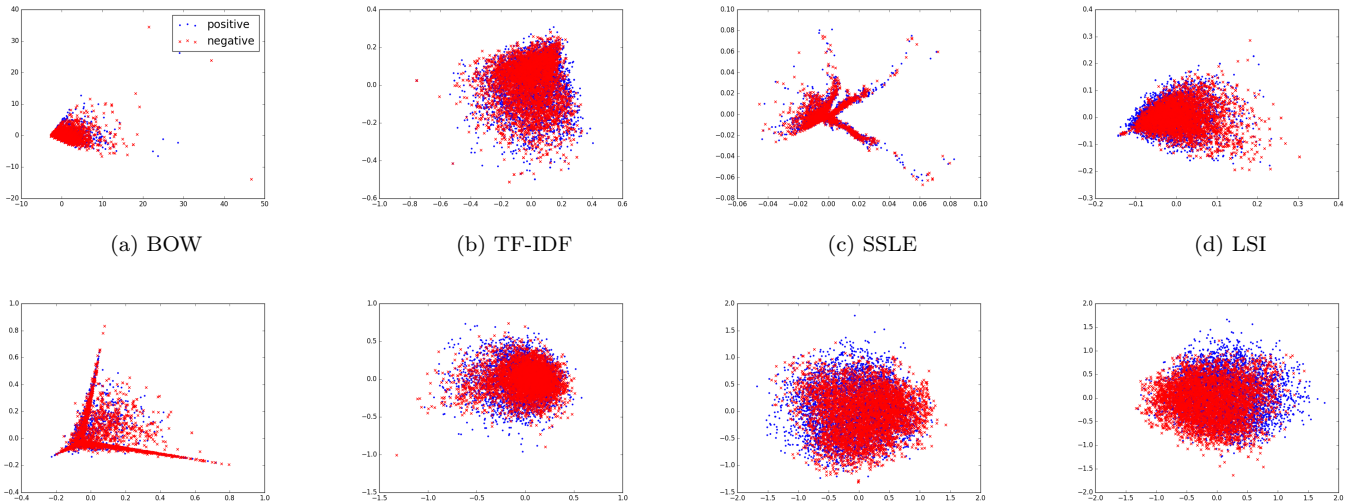
[4]Because SSLE needs to create and maintain a similarity matrix of documents whose size is $N \times N$, it can produce an out-of-memory problem. semi-DBOW only needs to maintain a neighborhood matrix whose size is $N \times k$.

4.4.3 Visualization

For sentiment visualization, we reduced the dimension using PCA for seven document representation models except for SSLE model, and all two-dimensional representations are plotted in 2D space. We used a small number of labeled documents (30 %) for our model and SSLE and visualized embeddings of unlabeled documents. In addition, for semi-DBOW, we fixed β to 0.05 and the number of neighbors to 10 based on the results of parameter analysis in Section 4.4.5. We compared eight visualization results for two-dimensional representations. The visualization results for Amazon review data of four categories are in Figures 4.3, 4.4, 4.5, and 4.6, where red points represent positive documents, and blue points represent negative documents.

For most categories, semi-DBOW showed good performance in separating the positive and negative sentiments of documents although we visualized the embeddings of unlabeled documents. In Figure 4.3, most models failed to distinguish the document sentiment, but Figure 4.3-(h) appears to be distinguishable. The similar patterns appeared in the other datasets more clearly than in the Book dataset.

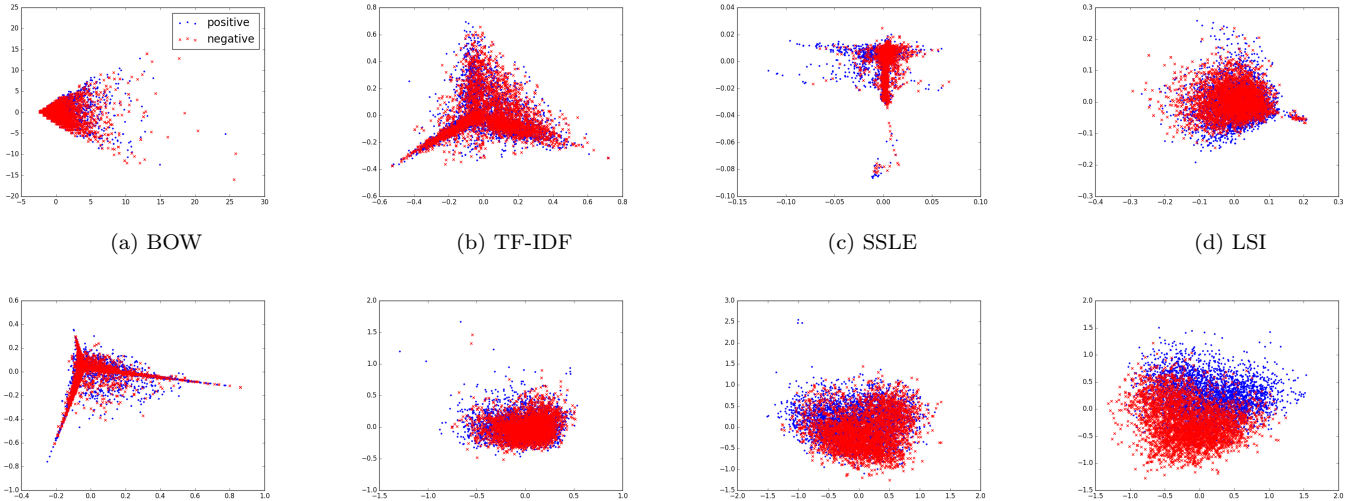
In Figures 4.4, 4.5 and 4.6, semi-DBOW can discriminate the sentiment classes well whereas another semi-supervised two-dimensional representation of SSLE using the partial sentiment information had difficulty in separation because we used only 30% of the sentiment labels of the whole data. We can see in Section 4.4.4 that the performances of SSLE are improved in 70% sentiment labels cases. Although LSI and TF-IDF models show good separation each in Electronics and Kitchen, our method is superior not only to these models even in Electronics and Kitchen but also shows stable performances. As a result, these visualization results imply that our document representations of unlabeled data reflect the difference of distributions on opposite sentiments well.



(a) BOW (b) TF-IDF (c) SSLE (d) LSI

(e) LDA (f) DM (g) DBOW (h) semi-DBOW

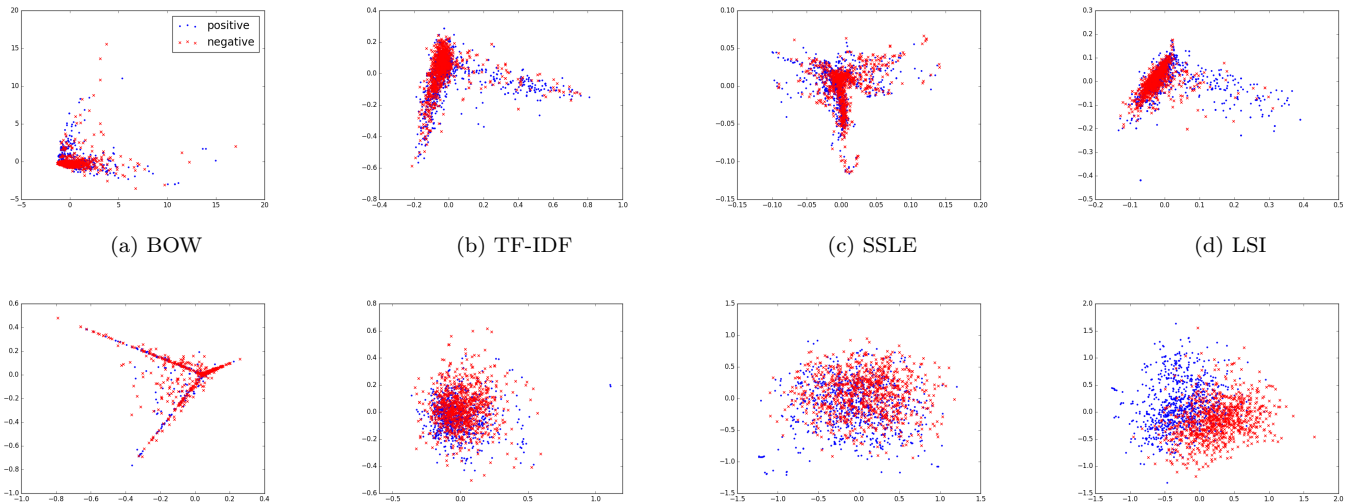
Figure 4.3: Two-dimensional scatter plots of Book dataset.



(a) BOW (b) TF-IDF (c) SSLE (d) LSI

(e) LDA (f) DM (g) DBOW (h) semi-DBOW

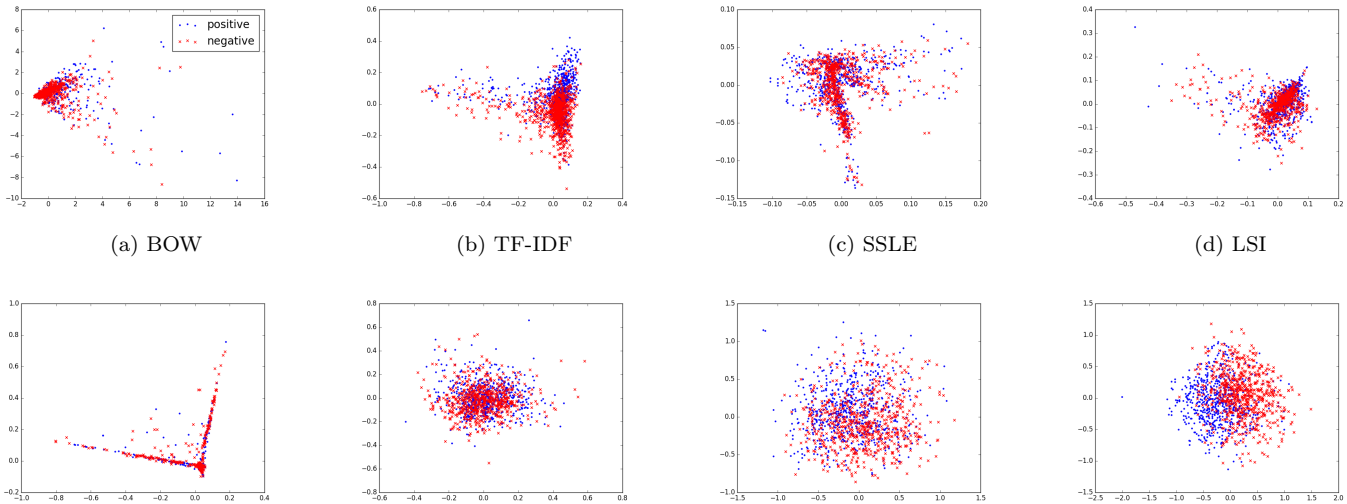
Figure 4.4: Two-dimensional scatter plots of DVD dataset.



(a) BOW (b) TF-IDF (c) SSLE (d) LSI

(e) LDA (f) DM (g) DBOW (h) semi-DBOW

Figure 4.5: Two-dimensional scatter plots of Electronics dataset



(a) BOW (b) TF-IDF (c) SSLE (d) LSI

(e) LDA (f) DM (g) DBOW (h) semi-DBOW

Figure 4.6: Two-dimensional scatter plots of Kitchen dataset.

4.4.4 Classification

We performed sentiment classification to compare the models quantitatively. Sentiment classification helps us to determine whether sentiment embeddings have effects on predicting the sentiments of documents. We performed the classification task on 200 dimensional document embeddings as well as two dimensional embeddings. Although the proposed method and SSLE used sentiment information partially, the comparison on sentiment classification is reasonable because the labels of training documents is necessary to construct classification model regardless of representation methods. We used a classification accuracy to compare all models and relative accuracy (RA) on DBOW and semi-DBOW to evaluate the effect of the label information, defined as follows:

$$RA = \frac{accuracy(semi - DBOW)}{accuracy(DBOW)}. \tag{4.6}$$

Table 4.3 shows the average prediction accuracy of the Amazon datasets with the standard deviation shown in parentheses and RA between DBOW and semi-DBOW, where the ratio of unlabeled documents is 0.7. In the table, the statistically significant best performing model on each case is starred, which is the same for the tables showing the experiment results of classification. We fixed the number of neighbors and β as the same as in section 4.4.3.

As expected, the average performance of semi-DBOW is superior to other models in the most cases. We found that the improvements of our model are significant when comparing the original Doc2vec models such as DM and DBOW, where RAs were larger than one in all cases. Semi-DBOW has particularly strong points in two-dimensional cases, which showed the superiority of our models quantitatively. Even if TF-IDF model showed comparable accuracies with our model in the two-dimensional case of Kitchen dataset, it exhibited poor performance for other datasets whereas our model had even and good prediction accuracies. While LSI and DBOW models

Classifier	Dim	BOW	TF-IDF	SSLE	LSI	LDA	DM	DBOW	semi-DBOW	RA
Book										
SVM	2	0.532	0.524	0.511	0.577	0.581	0.530	0.583	0.608*	1.04
		(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.00)	(0.00)	(0.01)	
	200	0.761	0.791		0.861	0.750	0.787	0.865	0.925*	1.06
		(0.01)	(0.00)		(0.00)	(0.01)	(0.01)	(0.00)	(0.00)	
LR	2	0.537	0.529	0.516	0.576	0.589	0.527	0.582	0.611*	1.04
		(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.00)	(0.01)	
	200	0.726	0.737		0.837	0.720	0.739	0.820	0.885*	1.07
		(0.01)	(0.00)		(0.00)	(0.01)	(0.01)	(0.01)	(0.00)	
DVD										
SVM	2	0.519	0.547	0.511	0.577	0.551	0.537	0.599	0.705 *	1.17
		(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	
	200	0.755	0.783		0.847	0.736	0.782	0.853	0.943*	1.1
		(0.01)	(0.01)		(0.00)	(0.01)	(0.00)	(0.00)	(0.00)	
LR	2	0.519	0.543	0.513	0.566	0.535	0.537	0.597	0.691*	1.15
		(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.00)	(0.01)	
	200	0.744	0.751		0.834	0.723	0.743	0.816	0.898*	1.09
		(0.01)	(0.01)		(0.00)	(0.01)	(0.01)	(0.01)	(0.00)	
Electronics										
SVM	2	0.514	0.571	0.528	0.603	0.499	0.614	0.675	0.825*	1.22
		(0.01)	(0.02)	(0.01)	(0.02)	(0.02)	(0.03)	(0.02)	(0.01)	
	200	0.761	0.782		0.831	0.711	0.745	0.822	0.916*	1.11
		(0.01)	(0.01)		(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	
LR	2	0.497	0.589	0.526	0.625	0.481	0.613	0.673	0.820*	1.21
		(0.03)	(0.02)	(0.01)	(0.02)	(0.02)	(0.02)	(0.02)	(0.01)	
	200	0.758	0.754		0.832	0.677	0.725	0.795	0.890*	1.11
		(0.01)	(0.02)		(0.01)	(0.02)	(0.01)	(0.01)	(0.01)	
Kitchen										
SVM	2	0.483	0.710*	0.585	0.603	0.494	0.535	0.617	0.709*	1.14
		(0.01)	(0.02)	(0.01)	(0.04)	(0.01)	(0.02)	(0.02)	(0.02)	
	200	0.742	0.762		0.847	0.617	0.683	0.776	0.863*	1.11
		(0.01)	(0.01)		(0.01)	(0.01)	(0.02)	(0.01)	(0.01)	
LR	2	0.491	0.709*	0.605	0.566	0.504	0.528	0.616	0.715*	1.11
		(0.02)	(0.01)	(0.01)	(0.04)	(0.01)	(0.02)	(0.01)	(0.01)	
	200	0.771	0.759		0.847*	0.723	0.686	0.780	0.853*	1.09
		(0.02)	(0.01)		(0.02)	(0.01)	(0.02)	(0.01)	(0.01)	

Table 4.3: Results of Amazon datasets in terms of the average accuracy for sentiment prediction with the unlabeled ratio 0.7

show the second or third best performances in most cases, LDA shows the worst performances in most cases. Moreover, we can also justify choosing the DBOW model as our base model from the fact that DBOW was more superior to DM in all cases. Unlike our expectation, however, SSLE gave the worst performance in large datasets, such as Book and DVD. This result might be because the performance of SSLE is easily affected by hyper-parameters such as the number of neighbors, weight of semi-supervised objective, or kernel parameter.

For Yelp datasets, we predicted the sentiment label from seven models except SSLE that suffers from memory problem on calculating similarity matrix of documents. Table 4.4 provides the average prediction accuracy and standard deviations of 10 runs on Yelp datasets. As Amazon datasets, semi-DBOW yields the best performances in all Yelp datasets, and it has significantly better prediction accuracy than the second best cases. While our model shows the aforementioned performances consistently, the second best model changes depending on dimension, dataset, and classifier. Moreover, RA in Table 4.4 shows that the amount of improvement of our model can sufficiently demonstrate the effectiveness of our model.

Classifier	Dim	BOW	TF-IDF	LSI	LDA	DM	DBOW	semi-DBOW	RA
2008									
SVM	2	0.527	0.639	0.644	0.616	0.54	0.634	0.852*	1.344
		(0.01)	(0.03)	(0.00)	(0.01)	(0.01)	(0.01)	(0.00)	
	200	0.727	0.774	0.856	0.696	0.628	0.64	0.925	1.445
		(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.02)	(0.00)	
LR	2	0.528	0.64	0.623	0.617	0.525	0.631	0.849	1.346
		(0.01)	(0.03)	(0.00)	(0.01)	(0.01)	(0.01)	(0.00)	
	200	0.774	0.77	0.870	0.736	0.582	0.665	0.92*	1.382
		(0.01)	(0.01)	(0.01)	(0.00)	(0.01)	(0.01)	(0.00)	
2010									
SVM	2	0.611	0.708	0.662	0.626	0.538	0.635	0.804*	1.266
		(0.01)	(0.01)	(0.00)	(0.00)	(0.01)	(0.00)	(0.00)	
	200	0.772	0.814	0.875	0.771	0.703	0.739	0.924*	1.25
		(0.01)	(0.01)	(0.00)	(0.01)	(0.01)	(0.00)	(0.01)	
LR	2	0.612	0.708	0.660	0.633	0.519	0.622	0.804*	1.292
		(0.00)	(0.01)	(0.00)	(0.00)	(0.01)	(0.01)	(0.01)	
	200	0.816	0.82	0.895	0.804	0.639	0.658	0.909*	1.381
		(0.00)	(0.01)	(0.00)	(0.00)	(0.01)	(0.01)	(0.01)	
2013									
SVM	2	0.638	0.743	0.669	0.625	0.554	0.651	0.833*	1.279
		(0.00)	(0.00)	(0.01)	(0.01)	(0.00)	(0.00)	(0.00)	
	200	0.802	0.839	0.885	0.781	0.804	0.854	0.947*	1.109
		(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	
LR	2	0.636	0.744	0.668	0.628	0.549	0.648	0.833*	1.285
		(0.00)	(0.00)	(0.00)	(0.01)	(0.00)	(0.00)	(0.00)	
	200	0.838	0.839	0.904	0.822	0.717	0.778	0.927*	1.191
		(0.00)	(0.00)	(0.00)	(0.00)	(0.01)	(0.00)	(0.00)	

Table 4.4: Results of Yelp datasets in terms of the average accuracy for sentiment prediction with unlabeled ratio 0.7

When we compare the results of Yelp datasets in Table 4.4 with the results of the

Amazon datasets in Table 4.3, we can find that the Yelp datasets tend to have higher RA and smaller standard deviations than Amazon datasets. Although in the Yelp datasets, the distributed representation models, such as DM and DBOW have inferior performances as compared to the count-based and topic models unlike in the Amazon datasets, semi-DBOW still shows superior performance. Therefore, we can conclude that our method can learn the effective document representation reflecting sentiment information and obtain stable performance regardless of classifier or dataset.

We performed additional experiments on the Amazon datasets having 70% labeled documents to verify the effect of using partial sentiment labels. Table 4.5 also shows that the performance of semi-DBOW is higher than other models in most datasets. In this case, the performances of SSLE were improved by increasing the ratio of labeled documents, which is illustrated in the Electronics and Kitchen.

From RAs in Table 4.3 and 4.5, we observed that our representations of all datasets improved the performance of sentiment classification models regardless of categories, the ratio of labeled documents, and classifiers. In most cases, RAs in Table 4.5 are higher than in Table 4.3. This result means that the more we use labeled documents, the larger the increase of prediction accuracy becomes. However, the differences of the amount of improvement are negligible when utilizing the full dimensional vector for prediction.

4.4.5 Parameter analysis

In previous sections, the experimental results on document representations of semi-DBOW were obtained from some fixed hyper-parameters. However, in this section, we inspect how the performance of our proposed algorithm is affected by hyper-parameters. The hyper-parameters considered include number of neighbors k, learning rate β, and the unlabeled document ratio R. For the analysis, we learned the document vectors with different combinations of hyper parameters, where $k \in \{3, 10, 30\}$,

Classifier	Dim	BOW	TF-IDF	SSLE	LSI	LDA	DM	DBOW	semi-DBOW	RA
Book										
SVM	2	0.526	0.511	0.528	0.573	0.585	0.528	0.572	0.616*	1.08
		(0.01)	(0.01)	(0.01)	(0.00)	(0.01)	(0.01)	(0.01)	(0.01)	
	200	0.761	0.791		0.863	0.772	0.792	0.870	0.924*	1.06
		(0.01)	(0.01)		(0.00)	(0.01)	(0.00)	(0.00)	(0.00)	
LR	2	0.529	0.510	0.529	0.574	0.591	0.509	0.569	0.618*	1.09
		(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	
	200	0.731	0.735		0.840	0.722	0.734	0.815	0.894*	1.1
		(0.01)	(0.01)		(0.00)	(0.01)	(0.00)	(0.01)	(0.01)	
DVD										
SVM	2	0.534	0.543	0.522	0.577	0.555	0.533	0.605	0.755*	1.25
		(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.03)	(0.01)	(0.01)	
	200	0.793	0.817		0.878	0.796	0.836	0.892	0.946*	1.06
		(0.00)	(0.00)		(0.01)	(0.01)	(0.01)	(0.00)	(0.00)	
LR	2	0.536	0.544	0.525	0.579	0.539	0.535	0.597	0.741*	1.24
		(0.01)	(0.01)	(0.01)	(0.01)	(0.01)	(0.02)	(0.01)	(0.01)	
	200	0.748	0.753		0.834	0.718	0.740	0.825	0.916*	1.11
		(0.01)	(0.01)		(0.01)	(0.01)	(0.01)	(0.00)	(0.00)	
Electronics										
SVM	2	0.489	0.544	0.640	0.595	0.551	0.588	0.619	0.797*	1.29
		(0.02)	(0.04)	(0.01)	(0.01)	(0.01)	(0.01)	(0.02)	(0.01)	
	200	0.719	0.759		0.863*	0.744	0.753	0.828	0.873*	1.05
		(0.01)	(0.01)		(0.01)	(0.01)	(0.02)	(0.02)	(0.02)	
LR	2	0.487	0.531	0.643	0.625	0.552	0.583	0.620	0.795*	1.28
		(0.03)	(0.05)	(0.01)	(0.02)	(0.02)	(0.02)	(0.02)	(0.01)	
	200	0.744	0.749		0.834	0.690	0.716	0.792	0.851*	1.07
		(0.01)	(0.01)		(0.01)	(0.01)	(0.01)	(0.02)	(0.00)	
Kitchen										
SVM	2	0.489	0.656	0.607	0.599	0.512	0.479	0.577	0.711*	1.23
		(0.01)	(0.02)	(0.01)	(0.02)	(0.02)	(0.01)	(0.02)	(0.02)	
	200	0.727	0.750		0.804	0.518	0.739	0.810	0.858*	1.06
		(0.02)	(0.01)		(0.01)	(0.03)	(0.01)	(0.01)	(0.02)	
LR	2	0.492	0.656	0.607	0.589	0.698	0.492	0.575	0.717*	1.28
		(0.02)	(0.02)	(0.01)	(0.03)	(0.01)	(0.00)	(0.02)	(0.02)	
	200	0.751	0.750		0.832*	0.637	0.695	0.778	0.831*	1.07
		(0.03)	(0.02)		(0.01)	(0.02)	(0.01)	(0.01)	(0.02)	

Table 4.5: Results of Amazon datasets in terms of the average accuracy for sentiment prediction with unlabeled ratio 0.3

$\beta \in \{0.01, 0.02, 0.03, 0.0, 50.1\}$, and $R \in \{0.3, 0.4, 0.5, 0.6, 0.7\}$. As same as the experiments in Section 4.4.4, we performed sentiment classification using SVM and LR for 10 times and then analyzed the effect of β and k. For the analysis, we used Electronics because it had the largest variance of classification results[5]. The results of the analysis of β are presented in Figure 4.7.

[5]For different R and β, the standard deviations of classification accuracies are 0.002 for Book, 0.003 for DVD, 0.019 for Electronics, and 0.009 for Kitchen. Electronics has much larger standard deviation than other datasets.

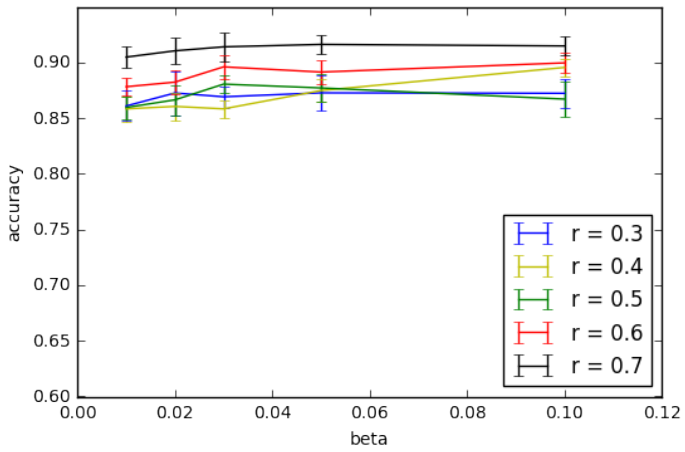


Figure 4.7: The prediction accuracy of different ratios of labeled documents according to the values of beta.

Regardless of R, the accuracy tends to increase as β increases, but the difference of accuracy between $\beta = 0.05$ and $\beta = 0.1$ is not significant in most cases. As a result, selecting the appropriate β values in the given range is not critical. Thus, in our experiments, we fixed β to 0.05.

Figure4.8 illustrates the influence of k on our model. The figure provides the accuracy for different ks, setting the β to 0.1 and unlabeled ratio R to 0.7. When k was equal to 10, our model had the best performances in all classification tasks, meaning that considering too many neighbors can be disastrous because doing so can result in excessive modification of the original data structure. However, using a too small k would result in limited impact on learning document embeddings. From these reasons, we set k to 10 in the experiments for sentiment visualization and classification.

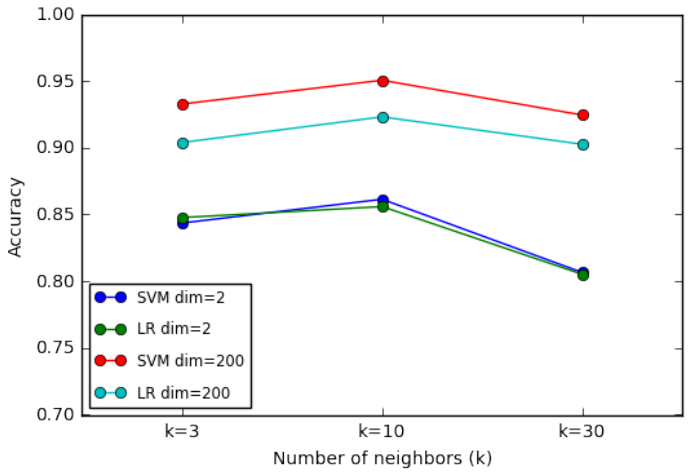


Figure 4.8: The prediction result of Electronics dataset for finding optimal number of neighbors k.

4.5 Chapter Summary

In the present chapter, we proposed distributed representation models of words and documents that address the lack of sentiment information. We obtained the effective continuous representations from the textual inputs which can be utilized to construct machine learning models for analyzing text data.

We attempted to develop a novel semi-supervised distributed representation learning algorithm that considered a sentiment-discriminative objective as well as the original DBOW objective of conserving the structure of original embedding space. While most existing semi-supervised or supervised document representation learning methods would learn document representation from the naive document representation, our method learned a document representation from scratch. In addition, we did not require additional information except the partial sentiment label of document corpus. Through plotting the document embeddings, we found that semi-DBOW can extract useful representations for sentiment analysis and opposite sentiment polarity labels

can be separated in the sentiment embedding space despite unlabeled documents. Additionally, the effectiveness of our method was verified by predicting sentiment label of unlabeled documents, where semi-DBOW not only exhibited the best performance in most datasets but also the stable results. The improvement of the original DBOW model was measured through RA, which is the accuracy ratio of semi-DBOW to DBOW, and RA in Yelp and Amazon datasets demonstrated our method was better in the sentiment classification tasks. Parameter analysis of learning rate β for the new objective illustrated it was not sensitive in certain ranges, and the appropriate number of neighbors was obtained regardless of classifiers.

Chapter 5

Domain-Adapted Distributed Representation

5.1 Chapter Overview

Domain adaptation assumes that the distributions of training (source) and test (target) data are similar but different, unlike the assumption in traditional machine learning. Domain adaptation approaches construct a predictive model across domains (source and target domains) by learning common features (Glorot et al., 2011; Gopalan et al., 2014), weighting instances (Gretton et al., 2009; Kanamori et al., 2009; Sugiyama et al., 2008), or matching training data to target data (Bollegala et al., 2016). One of the most important issues in the domain adaptation problem is obtaining good representation because the difference in distributions fundamentally originates from representations. Therefore, learning representations for domain adaptation is an interesting topic.

Representation learning methods for domain adaptation can be divided into semi-supervised and unsupervised methods according to whether target data are partially

labeled or unlabeled. Semi-supervised methods use partial target labels for training classifiers and learning newly transformed features from the original representations (Xiao & Guo, 2015; Yao et al., 2015; Daumé III et al., 2010). Many deep learning methods have been developed recently to simultaneously learn shared representations and predictive models; these methods have achieved state-of-the-art performance despite using only source labels (Ganin et al., 2016; Long et al., 2016; Bousmalis et al., 2016; Zellinger et al., 2017). While these unsupervised domain adaptation methods use source labels, some researchers have developed autoencoder-based methods that learn shared representations even without source labels (Glorot et al., 2011; M. Chen et al., 2012). These methods train stacked denoising autoencoder (sDAE) or marginalized stacked denoising autoencoder (mSDA) with source and target data without labels and obtain effective and generic features, which are invariant across domains in natural language processing (NLP) tasks. However, these models start from the numerical representations of texts instead of from textual inputs.

Therefore, we select distributed representation learning framework to obtain the numerical representations of texts from textual inputs. However, when representations from different domains are learned, distributed representation models suffer from domain-separation problem because the distributions of words are different. Bollegala et al. proposed an unsupervised cross-domain word representation method, based on distributed representation learning methods, which learns domain-specific word embeddings but has similar embeddings in regard of pivot words (Bollegala et al., 2015). However, because they focused only on learning word embeddings, document embeddings were not learned but obtained by the weighted sum of consisting words where source labels were used for learning the weights.

In the present study, we intend to develop a distributed representation learning method of documents for domain adaptation, which can capture semantic relationships, reduce the difference in the distributions of source and target documents, and

provide base document representation for other domain adaptation prediction models. The proposed method can not only learn document representations without any classification label but also from scratch (i.e., starting with the original text documents and not with primitive numerical representations such as BOW representations).

5.2 Representation Learning for Domain Adaptation

A model trained with source data is difficult to generalize to the target data when source and target distributions differ. In NLP applications, distribution depends on how documents or words are represented as numerical vectors, which means that learning representation is important in the domain adaptation problem. Therefore, most domain adaptation models aim to find new features that minimize the difference between the source and target domains from the numerical representations of words or documents. Glorot et al. (2011) used BOW representations as input and trained sDAE layer by layer to obtain hidden representations that can characterize documents across domains. M. Chen et al. (2012) learned new representations by using mSDA and overcame the limitations in Glorot et al. (2011), such as high computational cost and the high dimensionality problem caused by dictionary-based representation (M. Chen et al., 2012). Although mSDA is effective in domain adaptation for sentiment classification, it still requires the numerical representation of documents as an input (M. Chen et al., 2012).

State-of-the-art models that use deep learning techniques directly minimize domain divergence between domain-specific features and classification loss of representations in the source domain (Ganin et al., 2016; Bousmalis et al., 2016; Zellinger et al., 2017). These models are applicable to all application fields, where numerical vector representations are provided. Therefore, they also use BOW representations to represent textual input as numerical vectors when the sentiment classification task is

given. Ganin et al. developed the domain adversarial neural network (DANN) algorithm that applies the concept of adversarial training to extract common features from the source and target domains (Ganin et al., 2016). The extracted features should be discriminative to a given classification task and indiscriminative between the source and target domains, which can be achieved by using the gradient reversal layer in the domain classifier. The DANN algorithm minimizes H-divergence (Ben-David et al., 2010), which measures the distance between the source and target distributions. The DANN model trains new features from BOW and mSDA representations for a sentiment analysis dataset (Ganin et al., 2016). As shown in the result, DANN on mSDA demonstrates considerably superior performance than the original (BOW) representations, which implies that a more effective representation can affect the performance of domain adaptation. Therefore, we use our document representations as the input feature of DANN and train the DANN model to identify the effectiveness of our representations. In this study, we aim to develop an improved document representation learning method that can replace dictionary-based representation methods in NLP domain adaptation tasks. Distributed representation methods for words and documents have outperformed dictionary-based methods in many NLP tasks; thus, we introduce a new method based on these approaches.

Bollegala et al. (2015) proposed a distributed representation model of words for the domain adaptation task, where words are divided into pivots and non-pivots, and word embeddings for the source and target domains are trained separately (Bollegala et al., 2015). They used hinge loss instead of conditional probability (4.1) and considered only the relations of pivot and non-pivot words, thereby maximizing the prediction accuracy of non-pivot words in the fixed-length context of a pivot word in each domain

with the following equation:

$$\sum_{C\in\{\mathcal{S},\mathcal{T}\}}\sum_{d\in\mathcal{D}_C}\sum_{(w_p,w_n)\in d}\sum_{w_*\sim p_C(w)}\max\left(0,1-v^C_{w_p}\cdot v^C_{w_n}+v^C_{w_p}\cdot v^C_{w_*}\right) \tag{5.1}$$

where v^C_w is the word embedding of w for domain C of source $\mathcal{S}$ or target $\mathcal{T}$, and w_* is sampled from the 3/4th-powered marginal distribution of non-pivot words in domain C in Bollegala et al. (2015). The aforementioned objective is regularized by minimizing the differences in pivot word embeddings in the source and target domains. Our method is similar to this method in that it learns distributed representations for domain adaptation. However, our method simultaneously learns document and word representations and does not distinguish between source and target embeddings for words.

In summary, domain adaptation for NLP requires the effective embeddings of words and documents. Researchers have developed models learning common features, but these models start with numerical representations rather than textual input. Although the distributed representation framework has been effectively applied to learn word and document embeddings from textual input and has outperformed dictionary-based models, the latter is mostly used for domain adaptation. Therefore, we intend to develop a distributed representation method for documents that reduces domain divergence between source and target representations. Detailed examples and explanations are provided in Section 5.3.

5.3 Proposed Method

Most document representation models suffer from the domain separation problem in which the supports of document embeddings in the source and target domains do not coincide when simultaneously training document representations from different. For example, this problem can occur in dictionary-based models because the source and

target domains share only some of the words. If we use only common words, then document representations can lose a considerable amount of information. Although the Doc2Vec model can learn document embeddings that accurately reflect the relation between words, this model cannot prevent document embeddings from having different distributions across domains in the embedding space. In the current study, we focus on developing a document representation model based on the Doc2Vec model to address the domain separation problem.

Distributed representation models can significantly reduce computational complexity and yield effective representations by training with negative sampling (Mikolov, Sutskever, et al., 2013). The negative sampling method is inspired by the NCE method for the efficient learning of word embeddings (Mnih & Kavukcuoglu, 2013). Let p_{data} be the training data distribution and p_n be the noise distribution. To apply NCE, a new binary class variable C should be introduced for an auxiliary problem that distinguishes between real and noise data (Goodfellow et al., 2016), where a new model over an input word w_i, an output word w_j, and C can be specified as follows:

$$p_{joint}(w_j|w_i, C = 1) = p_\theta(w_j|w_i), \tag{5.2}$$

$$p_{joint}(w_j|w_i, C = 0) = p_n(w_j). \tag{5.3}$$

Similar distributions are constructed for the training data, where $p_{train}(w_j|w_i, C = 1) = p_{data}(w_j|w_i)$ and $p_{train}(w_j|w_i, C = 0) = p_n(w_j)$. Suppose that the negative examples from the noise distributions are k times more frequent than those in the real data ($p_{joint}(C = 1) = \frac{1}{k+1}$ and $p_{joint}(C = 0) = \frac{k}{k+1}$), and the input instances are independent of the class variable C. Then, the following logistic model can be

constructed:

$$p_{joint}(C=1|w_i,w_j;\theta) = \frac{p_\theta(w_j|w_i)}{p_\theta(w_j|w_i)+kp_n(w_j)} \tag{5.4}$$
$$= \sigma\left(\log p_\theta(w_j|w_i) - \log(kp_n(w_j))\right) = \sigma(\Delta s_\theta(w_j,w_i)),$$

where $\Delta s_\theta(w_j,w_i) = s_\theta(w_j,w_i) - \log\sum_k \exp s(w_k,w_i) - \log\left(kp_n(w_j)\right)$ which corrects (Mnih & Kavukcuoglu, 2013) and s_θ is given like in equation (4.2). The model can be fit by maximizing the log-posterior probability $\log p_w^i(C|w_j)$ averaged over data and noise distribution:

$$\mathbb{E}_{p_{data}(w_i)}\mathbb{E}_{p_{data}(w_j|w_i)}\left[\log\sigma(\Delta s_\theta(w_j,w_i))\right]$$
$$+ k\mathbb{E}_{p_{data}(w_i)}\mathbb{E}_{p_n(w_j)}\left[\log\left(1-\sigma(\Delta s_\theta(w_j,w_i))\right)\right]. \tag{5.5}$$

However, equation (5.5) requires intensive computations of the evaluation of the noise distribution p_n for an arbitrary point to calculate the objective function and its gradient. Mikolov, Sutskever, et al. (2013) proposed the negative sampling method that simplifies NCE by eliminating the evaluation of noise distributions $\Delta s_\theta(w_j,w_i) = \tilde{v}_j^\top v_i$ while maintaining their quality in Mikolov, Sutskever, et al. (2013). Therefore, they maximized equation (4.1) with the following conditional probability:

$$\log p(w_{t+j}|w_t) = \log\sigma(\tilde{v}_{w_{t+j}}^\top v_{w_t}) - \sum_{i=1}^{k}\mathbb{E}_{w_i\sim p_n(w)}[\log\sigma(-\tilde{v}_{w_i}^\top v_{w_t})] \tag{5.6}$$

This technique was also extended to train the distributed representations of documents in Le & Mikolov (2014). The DBOW model has the following objective:

$$\sum_{d\in\mathcal{D}}\sum_{w\in d}\log\sigma\left(\tilde{v}_w^\top v_d\right) + k\sum_{d\in\mathcal{D}}\mathbb{E}_{w'\sim p_n(w')}\left[\log\left(1-\sigma\left(\tilde{v}_{w'}^\top v_d\right)\right)\right], \tag{5.7}$$

where $\mathcal{D}$ is the training corpus. The expectation over data distribution is replaced by the training data. As indicated in Gutmann & Hyvärinen (2010), we can obtain good optimum quality by selecting noise distribution that is similar to the data distribution

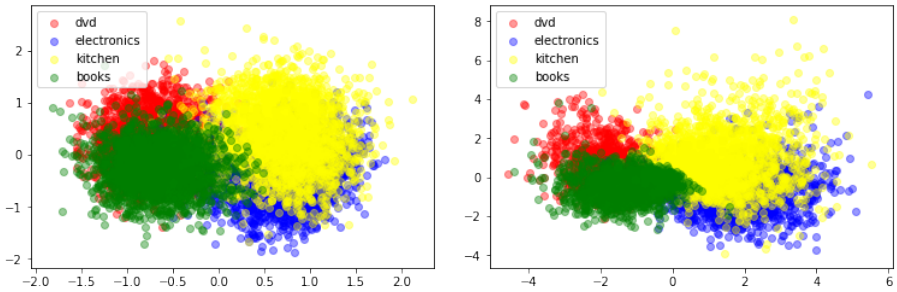


Figure 5.1: Document embeddings learned by DBOW and DM model from four different domains in Amazon review datasets: Book, DVD, Electronics and Kitchen.

in certain aspects. This fact was proven empirically in Mikolov, Sutskever, et al. (2013) because setting the 3/4th powered unigram distribution as the noise distribution produced good results. However, training a model with objective (5.7) from documents of multiple domains can separate document embeddings by domain because the model is learned to discriminate between model distribution and the common unigram-based noise distribution. Figure 5.1 illustrates document embeddings learned by the DBOW and DM models from four different domains in Amazon review datasets. As shown in the figure, the document embeddings of each domain are clustered rather than evenly spread in both distributed representation models. This representation property is inappropriate for domain adaptation tasks because of the high $\mathcal{H}$-divergence (Ben-David et al., 2010).

To solve this problem, we introduce a new auxiliary variable D for the domains, where $D = \mathcal{S}$ denotes the source domain, whereas $D = \mathcal{T}$ denotes the target domain. We assume that the domain variable D is independent of the class variable C. Then, we can construct a new model as follows:

$$p(w|d, C = 1, D) = p(w|d, C = 1) = p_{data}(w|d), \tag{5.8}$$

$$p(w|d, C = 0, D) = p(w|C = 0, D) = p_n(w|D). \tag{5.9}$$

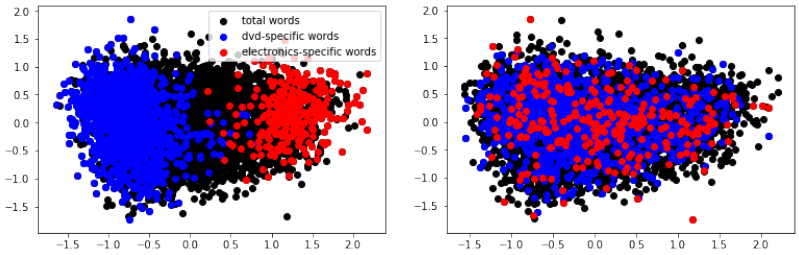
The second equality in equation (5.8) holds because the domain variable D is automatically determined given a document d. However, noise distribution is dependent on the domain variable D because this distribution is independent of the input variable d as shown in equation (5.3). We draw noise word samples and replace document samples with training documents based on the joint noise distribution $p_n(w, d)$.

$$\begin{aligned} p(w, d|C = 0) &= p(w, d, D = \mathcal{S}|C = 0) + p(w, d, D = \mathcal{T}|C = 0) \\ &= p(w|d, D = \mathcal{S}, C = 0)p(d|D = \mathcal{S})p(D = \mathcal{S}) + \\ &\quad p(w|d, D = \mathcal{T}, C = 0)p(d|D = \mathcal{T})p(D = \mathcal{T}) \\ &= p_n(w|D = \mathcal{S})p(d|D = \mathcal{S})p(D = \mathcal{S}) + \\ &\quad p_n(w|D = \mathcal{T})p(d|D = \mathcal{T})p(D = \mathcal{T}) \end{aligned}$$

Finally, we can construct the following objective of training document embeddings for domain adaptation by replacing the train distribution with the samples:

$$\sum_{d \in \mathcal{D}_\mathcal{S} \vee \mathcal{D}_\mathcal{T}} \sum_{w \in d} \log \sigma \left(\tilde{v}_w^\top v_d \right) + k \sum_{D \in \{\mathcal{S}, \mathcal{T}\}} \sum_{d \in \mathcal{D}_D} \mathbb{E}_{w' \sim p_n(w'|D)} \left[\log(1 - \sigma \left(\tilde{v}_{w'}^\top v_d \right)) \right], \quad (5.10)$$

where the 3/4th-powered unigram distribution of domain D is used for $p_n(w'|D)$. We alternatively maximize our objective (5.10) and the skip-gram objective (5.6) using the stochastic gradient descent method (Zinkevich et al., 2010). In equation (5.10), the domain-dependent noise distribution $p_n(w|D)$ can improve embedding quality by providing a more similar distribution to the data distribution $p_{data}(w|d)$ than the marginal noise distribution $p_n(w)$. Moreover, the domain-dependent noise distribution prevents the model distribution $p_\theta(w|d)$ of word and document embeddings from varying from the common marginal distribution of the two domains because the unigram-based noise distribution is used. When a document embedding is updated, words from the other domain cannot be sampled as noise words, and consequently, document embedding can be prevented from departing from the domain-specific words



(a) Proposed method without words (b) Proposed method with words

Figure 5.2: Word embeddings from two proposed methods where black dots represent common words, and red and blue dots represent domain-specific words.

of the other domain. Therefore, we can obtain effective document representations that are not domain-separable from objective (5.10).

We can decompose the proposed method into two cases depending on whether the skip-gram model also introduces the domain variable D. Several domain adaptive representation methods have attempted to separate domain-specific and domain-invariant features for word embeddings (Bollegala et al., 2016; Bousmalis et al., 2016), whereas others learn only domain-inseparable features (Blitzer et al., 2006; Bollegala et al., 2015). When using the domain-dependent noise distribution in equation (5.6), the embeddings of domain-specific words become separable even if the embeddings of the other words are shared. Meanwhile, the embeddings of all the words become inseparable when the domain variable D is not considered. We compared word embeddings from domain-dependent and domain-independent noise distributions by visualizing word embeddings to determine how the change in the noise distribution in equation (5.6) can influence the distribution of the embeddings of domain-specific words. We trained the representations based on the experimental design presented in Section 5.4.2. Figure 5.2 shows the word embeddings of DVD and Electronics reviews in the Amazon dataset. As expected, the embeddings of domain-specific words in Figure 5.2 (a) are separated depending on domains, whereas those in Figure 5.2 (b)

are not. This tendency is consistently observed in other pairs, the results of which are presented in Figure A.1 in the Appendix A. The resulting document embeddings can possess different attributes because they are affected by the word embeddings. In the following section, we verify the effectiveness of the document embeddings obtained using the proposed method through several experiments on real data.

5.4 Experimental Results

We evaluated our approach on the widely used Amazon dataset, which contains customer reviews with grades for purchased products. We learned document representations and performed visualization and classification tasks. For domain adaptation, we expect the appropriate document representations from different domains to not only overlap but also to be effective in training the predictor. Document representation will be evaluated from this perspective through visualization and classification.

5.4.1 Data description

In this experiment, we used 4 categories of the Amazon review dataset (Blitzer et al., 2007), namely, Book, DVD, Kitchen and Electronics. We produced 6 datasets for domain adaptation, which consisted of 6000 documents from 2 different categories:(Book, DVD),(Book, Electronics), (DVD, Electronics), (Kitchen, DVD), (Kitchen, Electronics), and (Kitchen, Book). For the classification task, we transformed the score between 1 and 5 to a sentiment label, where we regarded 1-2 points as negative sentiments and 4-5 points as positive sentiments. Moreover, our domain adaptation datasets were balanced for sentiment labels and domains. Table 5.1 shows the numbers of total words and domain-specific words for all pairs. As shown in the table, Book and DVD have various words, whereas Kitchen and Electronics have less domain-specific

words[1]. Consequently, Book and DVD share many common words, and Kitchen and Electronics have a relatively high proportion of common words (87.16%).

Data Sets A & B	Total Words	Domain-specific Words of A	Domain-specific Words of B
Book & DVD	5784	59	62
Book & Electronics	4682	651	184
DVD & Electronics	4797	758	196
Kitchen & Electronics	3435	218	223
Kitchen & Book	4675	170	760
Kitchen & DVD	4777	183	850

Table 5.1: Number of total words and number of domain-specific words in every pair (We only counted words that appears more than ten times in the documents)

5.4.2 Experimental design

While most document representation learning methods for domain adaptation are task-dependent because they use the source labels and learn new document representations from the numerical representations of documents and not from scratch (textual input), the proposed method is purely unsupervised (task-independent) and learns representations from scratch. Therefore, we compared the proposed method with different document representation methods, namely, BOW, TF-IDF, domain-dependent TF-IDF, DBOW, and DM. BOW, TF-IDF, and domain-dependent TF-IDF are dictionary-based models; hence, we selected the most frequent 5,000 vocabularies as the dictionary. In this study, we also proposed the domain-dependent TF-IDF method, which separately calculates the inverse document frequency of the source and target domains to reduce the impact of domain-specific words. For brevity, we denote each domain-dependent TF-IDF as each TF-IDF. We reduced the dimension of docu-

[1]In this experiment, we considered only the words that appear more than 10 times in the dataset without stemming. The number of domain-specific words counts the words that appear only in a specific category. For example, in the dataset of Book and DVD, 5784 words appear more than 10 times, 59 words appear only in Book reviews, and 62 words appear only in DVD reviews.

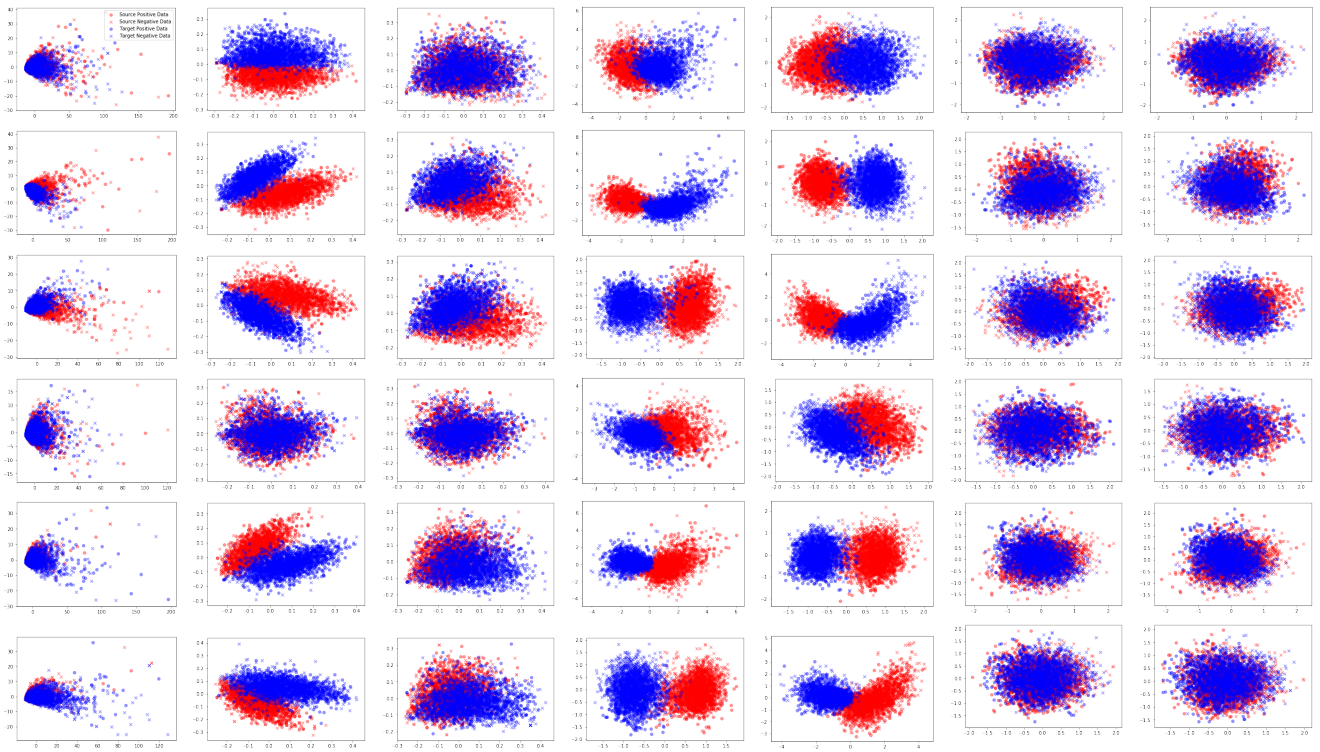
ment representations from dictionary-based models to 200 using principal component analysis (PCA) to address the curse of dimensionality. Unlike dictionary-based models, our methods and the distributed models (DBOW and DM) are required to set several hyper parameters to learn document representations. We used fixed hyper parameters (window = 3, minimum count = 10, negative count = 5, and dimension = 200) in our models and the comparative DBOW model because this setting provided consistent outputs in all the datasets. We proposed two distributed representation learning models of documents with and without changing the objective of input word embeddings, which are referred to hereafter as the proposed method with words and the proposed method without words, respectively. Therefore, we obtained the experimental results from seven different methods.

Representation learning was conducted with six combinations of four different domains in all the models because the role of the source and target datasets is not separated when learning representations in a purely unsupervised setting. Subsequently, we visualized and quantitatively measured the representations of the models after reducing the dimension to two via PCA to determine whether the source and target representations are mixed well. We used proxy A-distance (PAD), which measures how indistinguishable the data distributions are with respect to the domains. The A-distance is a measure of similarity among different distributions that was suggested in Ben-David et al. (2007). We practically measured the classification error ϵ of the support vector machine (SVM) classifier that was trained to discriminate between points sampled from different domains, where PAD is defined as $d_{PAD} = 2(1 - 2\epsilon)$. After randomly splitting the training and test data with a test ratio of 0.2, we trained the SVM classifier with the training data and calculated PAD with the remaining test data, where a linear kernel was used and the soft margin parameter was fixed to 10 as indicated in Ganin et al. (2016). We also compared the word embeddings from the two proposed methods by visualizing them.

We then applied document representations to sentimental analysis to determine if the representation can be transferred to sentimental labels. We conducted 12 cross-domain adaptation experiments with 4 different domains. We used the rbf SVM classifier with 0.01 as kernel parameter and 10 as margin parameter for classification, where the source data were used for training. Then, we tested the model directly on the target data. We compared classification accuracies in the case of 2D and 200D representations. Finally, we trained the new common embeddings and sentimental classifier by applying a deep learning-based domain adaptation model to the obtained document representations. We used the DANN model because it is simple but comparable with other state-of-the-art algorithms (Ganin et al., 2016). We also examined the convergence of the target test error and the classification accuracies to train the DANN model.

5.4.3 Visualization

We visualized document representations from five models after applying PCA. Figure 5.3 shows the visualization results. Each row represents a pair of two domains (A and B). The red circle corresponds to a positive A document, the red cross corresponds to a negative A document, and the blue color is for the B domain. The support regions of the domains in the Book and Electronics and DVD and Electronics pairs are separated in the visualization results of BOW. The representations of each TF-IDF have improved compared with the original TF-IDF because they are more overlapped even if the Book and Electronics and DVD and Electronics pairs are slightly separable. In the Kitchen and Electronics pair, the TF-IDF is extremely similar to each TF-IDF, which implies that the word distributions of the domains are similar to one another. The proposed method was consistent in adequately mixing the document representations from two different domains in all the cases, whereas the representations of DBOW were separated according to domain despite having similar objective functions, except



(a) BOW (b) TF-IDF (c) each TF-IDF (d) DBOW (e) DM (f) Proposed method (g) Proposed w/ words

Figure 5.3: Two dimensional plots of document representations of Amazon dataset. Each row contains a pair of two categories (A and B) in the following order: Book and DVD, Book and Electronics, DVD and Electronics, Kitchen and Electronics, Kitchen and Book, Kitchen and DVD.

Source & Target	BOW	TFIDF	each TF-IDF	DM	DBOW	Proposed method	Proposed w/ words
B & D	0.0364	1.7480	0.040	1.5440	1.5560	0.2480	0.2160
B & E	0.5640	1.8396	0.9596	1.8636	1.9116	0.2360	0.1760
D & E	0.6360	1.7280	1.036	1.8476	1.8676	0.0200	0.0200
K & E	-0.0324	0.1720	0.1080	1.5480	1.5720	0.0520	0.0720
K & B	0.5480	1.8076	0.8476	1.8556	1.9456	0.5440	0.3836
K & D	0.5560	1.6160	0.4756	1.5480	1.8956	0.0520	0.2440

Table 5.2: Proxy A-distance of 2-dimensional data

for the noise distribution of the output words.

Moreover, we used PAD to quantitatively demonstrate that the proposed method results in the indistinguishable distributions of document embeddings with respect to domains. On the basis of the definition, the distributions of the source and target representations are similar when PAD is low. The PADs of 2D representations are provided in Table 5.2.

Table 5.2 shows that the proposed method consistently exhibits low PAD measures regardless of the dataset pairs. As mentioned in Section 5.4.1, BOW and each TF-IDF can have low PADs on the Book and DVD and Kitchen and Electronics pairs because these pairs have a relatively high proportion of common words. The DBOW and TF-IDF methods have high PADs for all the dataset pairs, thereby indicating that these methods can make document distributions separable to consider word distribution based on the total document corpus. From these visualizations and the PAD results, we can conclude that our novel domain-dependent noise distribution of words and documents is effective for learning document embeddings with similar distributions across domains.

5.4.4 Sentiment classification

For domain adaptation, we require document representations that overlap across domains in terms of unsupervised learning, regardless of the specific NLP task. However, these representations should also conceive effective information about texts. In this section, we evaluate document representations through a sentiment classification task. Experiments were conducted on the 2D and 200D representations presented in Section 5.4.3. We measured the performance of the SVM classifier, which was trained on the source data and tested on the target data for 12 source and target pairs, to determine if the representation methods had learned the document embeddings that were informative to the sentiment invariant across domains without applying any covariate shift techniques to the classifier. Table 5.3 presents the domain adaptation accuracies of the Amazon review datasets in 2D data.

In this table, the proposed methods consistently exhibits high accuracies in all the experimental pairs compared with other word representations. The proposed methods achieve comparative accuracies, where the proposed method with words exhibits slightly superior performance. We can infer that BOW, TF-IDF, each TF-IDF, and the DM model experience difficulty in discriminating among the sentiments of document embeddings because their accuracies are nearly 50% in binary classification. The DBOW representation showed fine results in a few pairs but did not perform consistently in all the pairs. From the aforementioned results, we can conclude that our suggested methods mix source and target distributions, thereby maintaining their sentimental information. Accordingly, we can determine that the proposed methods extract sentiment transferable features among different domains.

We also performed domain adaptation sentimental analysis by using the original representations of the document representation model. We used PCA to document the embeddings of the BOW, TF-IDF, and each TFIDF models to reduce the dimension to

Source → Target	BOW	TF-IDF	each TF-IDF	DM	DBOW	Proposed method	Proposed w/ words
B → D	51.95 %	52.80 %	58.19 %	50.80 %	52.52 %	61.04 %	**61.60 %**
B → E	52.12 %	56.47 %	56.59 %	49.84 %	51.60 %	**61.19 %**	60.24 %
B → K	50.56 %	56.47 %	52.20 %	50.16 %	54.07 %	63.95 %	**65.12 %**
D → B	50.31 %	50.96 %	58.96 %	52.40 %	54.44 %	61.19 %	**62.12 %**
D → E	53.23 %	54.60 %	56.72 %	49.39 %	60.16 %	62.96 %	**65.12 %**
D → K	52.48 %	55.27 %	53.23 %	51.31 %	62.80 %	**67.20 %**	66.15 %
E → B	51.59 %	54.64 %	55.84 %	52.52 %	53.83 %	69.64 %	**70.04 %**
E → D	51.55 %	53.15 %	56.84 %	49.91 %	71.79 %	70.44 %	**71.84 %**
E → K	53.55 %	56.20 %	56.75 %	61.39 %	71.39 %	71.40 %	**72.96 %**
K → B	50.60 %	54.52 %	52.28 %	50.80 %	68.27 %	75.20 %	**75.60 %**
K → D	49.91 %	53.35 %	52.92 %	51.55 %	**72.16 %**	69.27 %	70.24 %
K → E	52.43 %	56.04 %	55.95 %	60.24 %	67.52 %	71.16 %	**72.19 %**

Table 5.3: Domain Adaptation accuracy of Amazon review datasets in two dimensional data

200 to have the same dimension to distributed representations because the dimensions of the original representations were too high (5000). The results of the sentimental analysis of the 200D document representations are illustrated in Figure 5.4.

This figure shows that our suggested models outperformed the other representations in all the dataset pairs. In particular, the proposed method with words achieved better results in more experimental pairs than the model without words. The Doc2vec models exhibited similar patterns, but the DBOW models demonstrated slightly better performance than DM. TF-IDF and each TF-IDF also obtained similar results, whereas the BOW models presented the worst performance in eight experiments. Many common words exist between the Book and DVD and Kitchen and Electronics pairs. All the results showed high performance when the word distributions in the source and target domains were similar, such as (Book $\rightarrow$ DVD), (Electronics $\rightarrow$ Kitchen), and (Kitchen $\rightarrow$ Electronics), and our suggested methods did not demonstrate considerable performance improvement compared with the other methods. By contrast, our methods significantly outperformed the other methods when the word distributions differed, such as (DVD $\rightarrow$ Electronics), (DVD $\rightarrow$ Kitchen), and (Kitchen $\rightarrow$ Book).

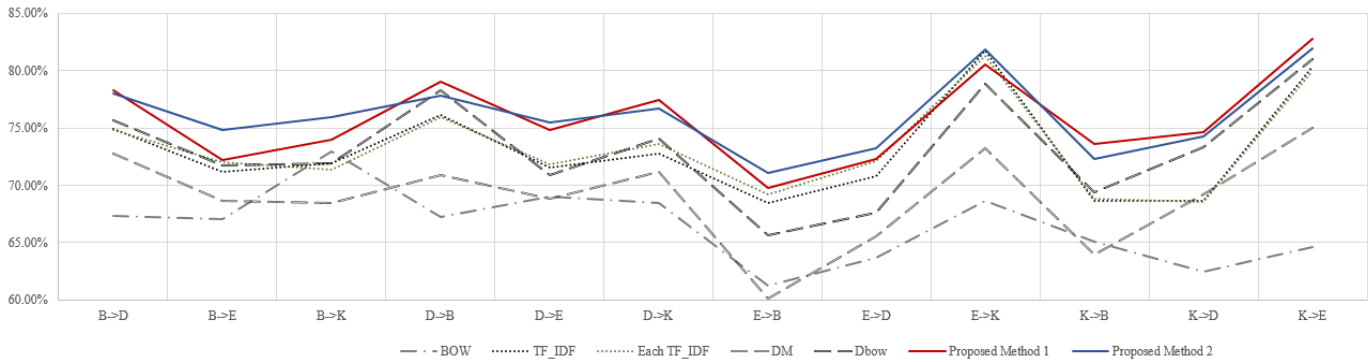


Figure 5.4: Sentimental analysis of document representations.

5.4.5 Application to Domain Adversarial Neural Network

As previously mentioned, the document embeddings from the proposed methods can be applied to other domain adaptation methods starting from the numerical vectors because our methods can learn numerical representations from textual input. To identify this useful feature, we also trained the suggested new common embeddings and sentimental classifier by applying the DANN model. We wanted to ascertain that the suggested text representations could improve the accuracy of the sentimental classifier when applied to the latest deep learning-based domain adaptation model. In the DANN model, we used two feature extraction layers (input dimension $\rightarrow$ 200 $\rightarrow$ 100), which had fully connected linear layers with the ReLu activation function. In addition, we set two layers (100 $\rightarrow$ 100 $\rightarrow$ 1) with the sigmoid function for the domain and sentimental classifiers. We stopped training the model when the accuracy of the source data approached one.

The results of the DANN application are shown in Figure 5.5, where the proposed methods apparently had high performance in all the pairs. In some cases, however, such as (DVD $\rightarrow$ Book), (DVD $\rightarrow$ Kitchen), (Electronics $\rightarrow$ DVD), and (Kitchen $\rightarrow$ Electronics), DBOW, each TF-IDF, and TF-IDF achieved slightly better results. One possible reason for this finding is that the DANN model attempts to obtain new common embeddings that are indiscriminate with respect to the source and target domains, so it is able to offset the advantageous properties of our representations. Nevertheless, the proposed method performed the best for the eight experimental pairs and consistently exhibited good performance.

We also visualized accuracy by epoch in Figure 5.6 to examine the convergence aspect of adversarial training. We showed only the cases where the proposed methods demonstrated inferior performance, i.e.,(DVD $\rightarrow$ Book),(DVD $\rightarrow$ Kitchen), and (Kitchen $\rightarrow$ Electronics); and the remaining cases are presented in Figure A.2 in Ap-

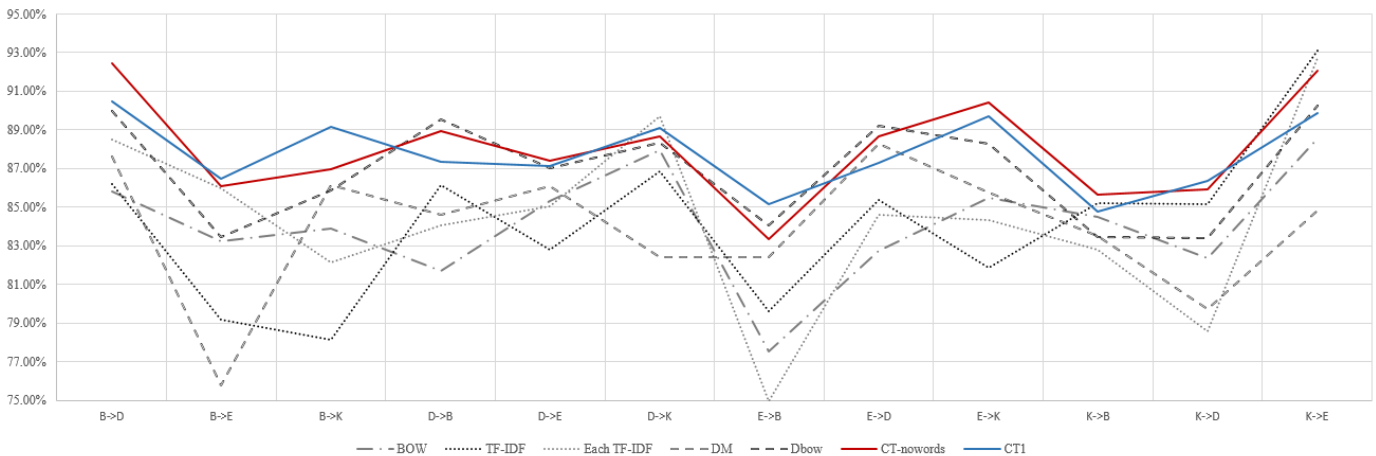
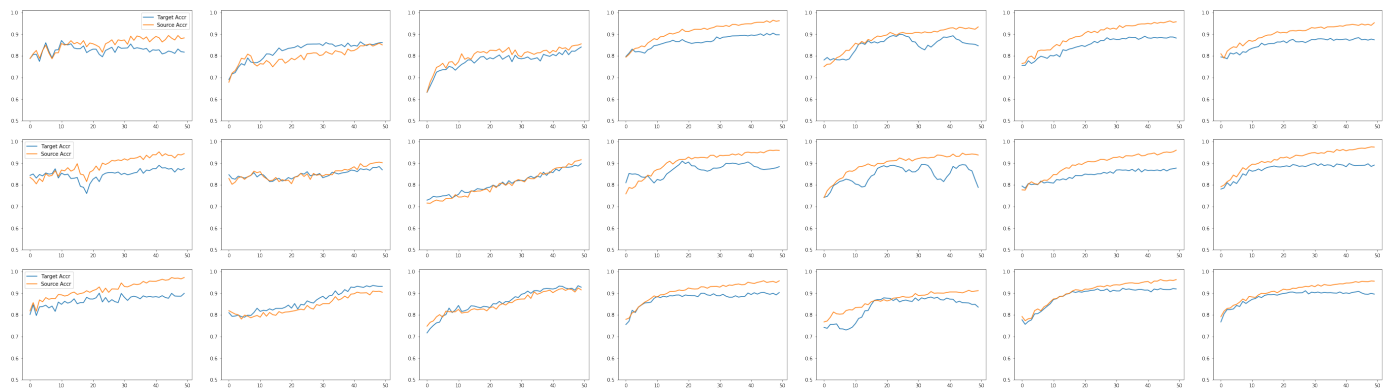


Figure 5.5: Sentimental analysis of document representations by using DANN model.

pendix A. Figure 5.6 shows that both models exhibit consistent increasing trend compared with the other methods. BOW, TF-IDF, and each TF-IDF fluctuated accuracy per epoch despite the decaying learning rate in training. In the DBOW and DM models, the accuracy of the source data consistently increased, whereas the accuracy of the target data unstably changed. Although our methods exhibited slightly lower performance in these datasets, their target accuracy followed the source accuracy and stably increased. The target training error of the proposed methods was insensitive to the feature extraction phase in DANN because our document embeddings of the source and target data were already similar. However, the target accuracy of the other methods depends considerably on the feature extraction of DANN, and therefore, the corresponding graph fluctuates through epochs. We can conclude that although the performance of our proposed model is not the best in all the experimental pairs, our methods not only demonstrated robust performance but also helped adversarial training to converge stably.

5.5 Chapter Summary

In this chapter, we proposed a novel distributed representation learning method of words and documents for the domain adaptation by exploring the negative sampling method and utilizing useful properties from this method. Our model can learn domain-indiscriminate document features in a purely unsupervised manner from textual input by using sophisticated noise distribution. We showed that the proposed method mixes source and target data by visualizing data distribution in 2D and calculating PAD measures. Moreover, we conducted a sentimental domain adaptation classification task, wherein the document embeddings of the source domain were used to train the SVM classifier and then the trained classifier was directly applied to target embeddings. The proposed models outperformed other comparative methods in



(a) BOW (b) TF-IDF (c) each TF-IDF (d) DBOW (e) DM (f) Proposed method (g) Proposed with words

Figure 5.6: Accuracy by epoch on document representations of Amazon dataset in experiments (DVD → Book), (DVD → Kitchen), and (Kitchen → Electronics). Orange line and blue line refer to source training accuracy and target training accuracy respectively.

2D and original embeddings. Finally, we showed that our method can be efficiently applied to one of the latest deep learning algorithms, i.e., DANN.

Chapter 6

Conclusion

6.1 Summary

In data analytics, machine learning and deep learning models have been widely and effectively used, but these models are highly dependent on the choice of data representation. Density-based representation learning extracts the useful information from data by learning the input density implicitly or explicitly. Also, textual inputs should be first represented as the numerical vectors and distributed representation of textual inputs should be improved to be effectively applied to sentiment analysis and domain adaptation tasks. In this dissertation, we focused on density-based representation learning and distributed representation learning of words and documents.

In chapter 2 and 3, we learned the input density using DAE and SVDD models, developed a nonlinear projection algorithm and inductive ensemble clustering method by constructing a dynamical system, and proposed low-density regularization of DAE using kernel radius function of SVDD model. In chapter 4 and 5, distributed representation learning methods for NLP were developed to disentangle the difficulty of

reflecting sentiment information and the domain separation problem. Contributions of this thesis are as follows:

- We demonstrated that the manifold could be learned from DAE model, and the dynamical system using the reconstruction function could asymptotically converge to the manifold by newly introducing *distribution on manifold*. We also proved that on some mild conditions the attracting equilibrium manifold of dynamical system using the score function of corrupted data is completely stable. We also performed visualization and predictive analysis of real-world image data to show the effectiveness of nonlinear projection algorithm based on the induced dynamical system.

- We proposed new inductive ensemble clustering method and regularization methods for DAE model using kernel radius function of SVDD model. We can obtain more robust clustering results as well as capture low-density region of the input space from DAE model by using the energy of SVDD model.

- We developed semi-supervised distributed representation learning for sentiment analysis to reflect sentiment information with preserving the local structure of distributed representation. This model can overcome the weakness of distributed representations on reflecting sentiment relationship.

- We also proposed domain adapted representation learning of words and documents that introduced domain-dependent noise distribution. Experimental results for real-world datasets showed that our methods had desirable influences on document embeddings.

6.2 Future Work

Although representation learning is getting more and more research attention with the development of machine learning and deep learning algorithm in data analytics, it still has the potential for improvement. This dissertation can provide some future work for developing an improved density-based representation learning method or applying proposed methods to various problems in data analytics.

- Our nonlinear projection algorithm can be generally applied to pre-processing stages in machine learning algorithms because this analysis helps in capturing the complex-shaped input data structures or reducing noises. Furthermore, other variants of autoencoder, such as convolutional DAE and contractive autoencoder, can also be exploited to train the input density. If they capture the structure of the input density better, the performance of the projection will be also improved over our present algorithm, thereby requiring further investigation.

- We combined the energy from SVDD model with DAE model to pull up the energy outside the support region in the input space. However, although the effect of low-density separation regularization was demonstrated by the illustrative examples, the induced dynamical system of the trained reconstruction function is not yet verified in terms of convergence and stability. Thus, further research is required from the theoretical point of view.

- Because our semi-supervised distributed representation learning method attempted to learn document embeddings that contain various useful distributional information of documents in addition to sentiment information by preserving the original embedding structure, this method can be applied effectively to diverse fields such as finance and marketing, which can require topics or key

words. In addition, although we focused on sentiment analysis in this paper, our method may reflect other information or even represent other sequential data.

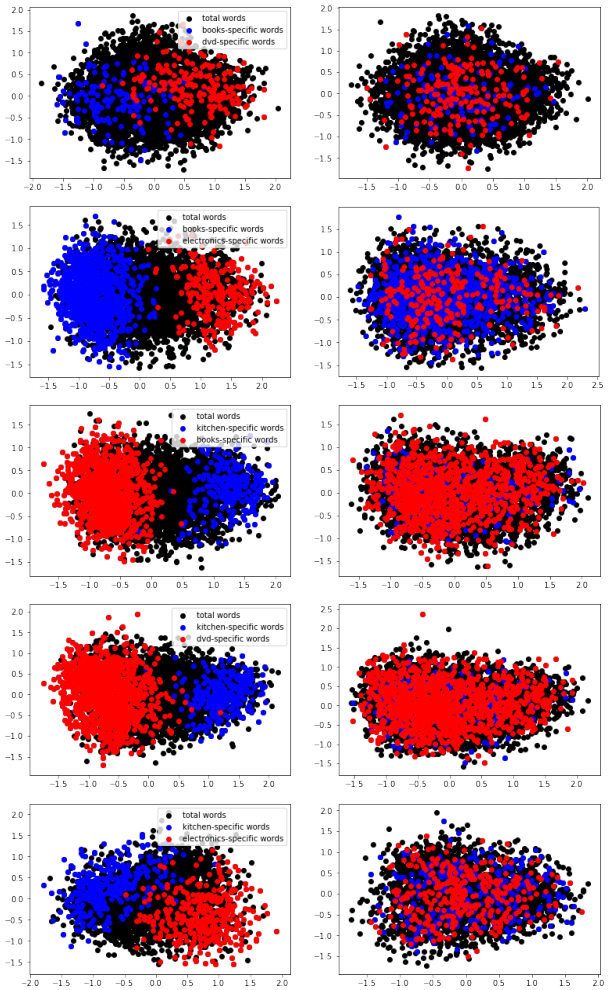
- We expect that our domain adapted distributed representation learning method can be extended to domain adaptation methods for various NLP tasks including the latest deep learning based domain adaptation models.

Appendix A

Domain Adaptation Figures

Figure A.1 shows the word embeddings for the pairs except for DVD and Electronics pair in Figure 5.2. All pairs have the domain-separable word embeddings for the domain-independent noise distribution and the domain-inseparable word embeddings for the domain-dependent noise distribution.

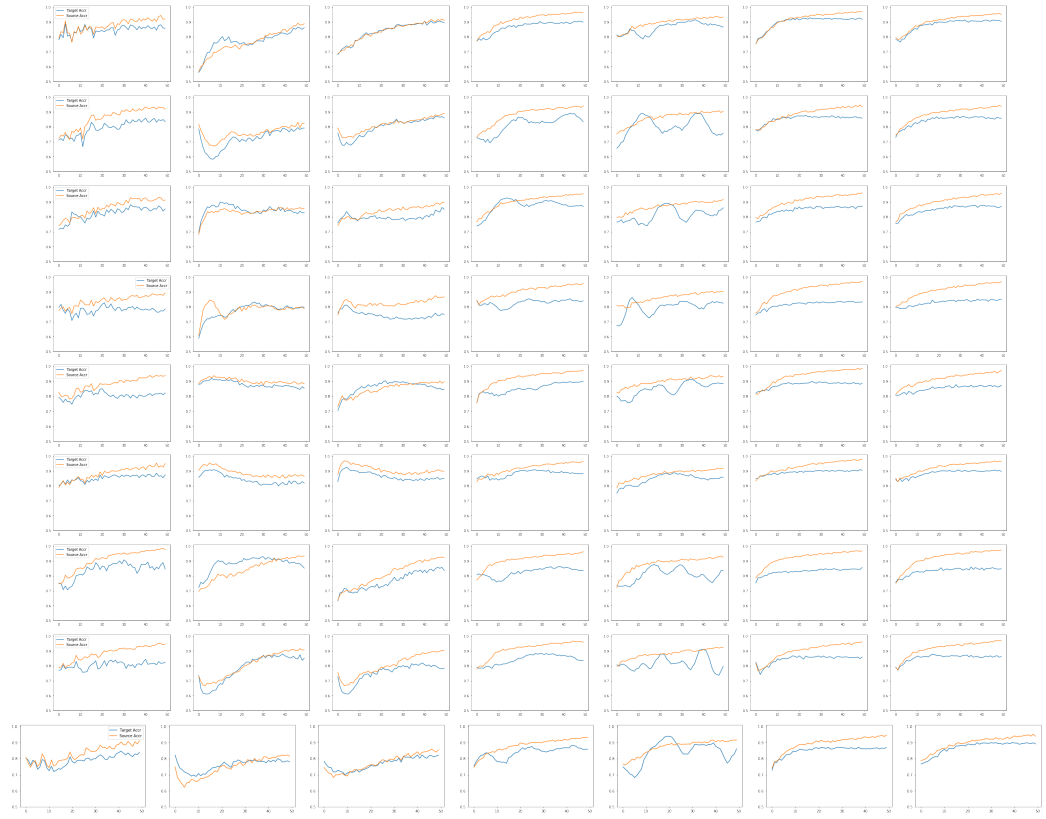
Figure A.2 shows the accuracies by epoch on the remaining pairs except for the pairs in Figure 5.6 when training DANN. We can find that two our proposed methods have the stably increasing patterns comparing other methods as in Figure 5.6.



(a) Proposed method without words

(b) Proposed method with words

Figure A.1: Word embeddings from two proposed methods where black dots represent common words, and red and blue dots represent domain-specific words.



(a) BOW (b) TF-IDF (c) each TF-IDF (d) DBOW (e) DM (f) Proposed method (g) Proposed w/ words

Figure A.2: Accuracy by epoch on document representations of Amazon dataset in experiments (Book → DVD), (Book → Electronics), (DVD → Electronics) (Electronics → Book), (Electronics → DVD), (Electronics → Kitchen), (Kitchen → Electronics) and (Book → Kitchen). Orange line and blue line refer to source training accuracy and target training accuracy respectively.

Bibliography

Alain, G., & Bengio, Y. (2014). What regularized auto-encoders learn from the data-generating distribution. *The Journal of Machine Learning Research*, *15*(1), 3563–3593.

Andrzejewski, D., & Zhu, X. (2009). Latent dirichlet allocation with topic-in-set knowledge. In *Proceedings of the naacl hlt 2009 workshop on semi-supervised learning for natural language processing* (pp. 43–48).

Asuncion, A., & Newman, D. (2007). *Uci machine learning repository.*

Becker, S., & Hinton, G. E. (1992). Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, *355*(6356), 161–163.

Belkin, M., & Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, *15*(6), 1373–1396.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., & Vaughan, J. W. (2010). A theory of learning from different domains. *Machine learning*, *79*(1), 151–175.

Ben-David, S., Blitzer, J., Crammer, K., & Pereira, F. (2007). Analysis of representations for domain adaptation. In *Advances in neural information processing systems* (pp. 137–144).

Ben-David, S., Lu, T., Pál, D., & Sotáková, M. (2009). Learning low density separators. In *International conference on artificial intelligence and statistics* (pp. 25–32).

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, *35*(8), 1798–1828.

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, *3*(Feb), 1137–1155.

Ben-Hur, A., Horn, D., Siegelmann, H. T., & Vapnik, V. (2001). Support vector clustering. *Journal of machine learning research*, *2*(Dec), 125–137.

Bickel, S., Brückner, M., & Scheffer, T. (2009). Discriminative learning under covariate shift. *Journal of Machine Learning Research*, *10*(Sep), 2137–2155.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, *3*(Jan), 993–1022.

Blitzer, J., Dredze, M., & Pereira, F. (2007). Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Acl* (Vol. 7, pp. 440–447).

Blitzer, J., McDonald, R., & Pereira, F. (2006). Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing* (pp. 120–128).

Bojchevski, A. (2017). *Learning by denoising part 2. connection between data distribution and denoising function.* Retrieved 2017-01-29, from `https://thecuriousaicompany.com/connection-to-g/#comment-15`

Bollegala, D., Maehara, T., & Kawarabayashi, K.-i. (2015). Unsupervised cross-domain word representation learning. *arXiv preprint arXiv:1505.07184*.

Bollegala, D., Mu, T., & Goulermas, J. Y. (2016). Cross-domain sentiment classification using sentiment sensitive embeddings. *IEEE Transactions on Knowledge and Data Engineering*, *28*(2), 398–410.

Boureau, Y.-L., Chopra, S., & LeCun, Y. (2007). A unified energy-based framework for unsupervised learning. In *Artificial intelligence and statistics* (pp. 371–379).

Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., & Erhan, D. (2016). Domain separation networks. In *Advances in neural information processing systems* (pp. 343–351).

Boyd-Graber, J., & Resnik, P. (2010). Holistic sentiment analysis across languages: Multilingual supervised latent dirichlet allocation. In *Proceedings of the 2010 conference on empirical methods in natural language processing* (pp. 45–55).

Bracewell, R. (1999). The sifting property. *The Fourier Transform and Its Applications, 3rd ed. New York, McGraw-Hill*, 74–77.

Chapelle, O., & Zien, A. (2005). Semi-supervised classification by low density separation. In *Aistats* (pp. 57–64).

Chen, L.-S., Liu, C.-H., & Chiu, H.-J. (2011). A neural network based approach for sentiment classification in the blogosphere. *Journal of Informetrics*, *5*(2), 313–322.

Chen, M., Xu, Z., Weinberger, K., & Sha, F. (2012). Marginalized denoising autoencoders for domain adaptation. *arXiv preprint arXiv:1206.4683*.

Chollet, F. (2015). *Keras.* `https://github.com/fchollet/keras`. GitHub.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, *12*(Aug), 2493–2537.

Dai, A. M., Olah, C., & Le, Q. V. (2015). Document embedding with paragraph vectors. *arXiv preprint arXiv:1507.07998*.

Dai, Z., Almahairi, A., Bachman, P., Hovy, E., & Courville, A. (2017). Calibrating energy-based generative adversarial networks. *arXiv preprint arXiv:1702.01691*.

Daumé III, H., Kumar, A., & Saha, A. (2010). Frustratingly easy semi-supervised domain adaptation. In *Proceedings of the 2010 workshop on domain adaptation for natural language processing* (pp. 53–59).

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, *41*(6), 391.

Fred, A. L., & Jain, A. K. (2005). Combining multiple clusterings using evidence accumulation. *IEEE transactions on pattern analysis and machine intelligence*, *27*(6), 835–850.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., ... Lempitsky, V. (2016). Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, *17*(59), 1–35.

Glorot, X., Bordes, A., & Bengio, Y. (2011). Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning (icml-11)* (pp. 513–520).

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Gopalan, R., Li, R., & Chellappa, R. (2014). Unsupervised adaptation across domain shifts by generating intermediate data representations. *IEEE transactions on pattern analysis and machine intelligence*, *36*(11), 2288–2302.

Gretton, A., Smola, A. J., Huang, J., Schmittfull, M., Borgwardt, K. M., & Schölkopf, B. (2009). *Covariate shift by kernel mean matching*. MIT press.

Gupta, V., & Lehal, G. S. (2009). A survey of text mining techniques and applications. *Journal of emerging technologies in web intelligence*, *1*(1), 60–76.

Gutmann, M., & Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Aistats* (Vol. 1, p. 6).

Harris, Z. S. (1954). Distributional structure. *Word*, *10*(2-3), 146–162.

Hinton, G. E., & Salakhutdinov, R. R. (2006a). Reducing the dimensionality of data with neural networks. *science*, *313*(5786), 504–507.

Hinton, G. E., & Salakhutdinov, R. R. (2006b). Reducing the dimensionality of data with neural networks. *science*, *313*(5786), 504–507.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of classification*, *2*(1), 193–218.

Jung, K.-H., Kim, N., & Lee, J. (2011). Dynamic pattern denoising method using multi-basin system with kernels. *Pattern Recognition*, *44*(8), 1698–1707.

Jung, K.-H., Lee, D., & Lee, J. (2010). Fast support-based clustering method for large-scale problems. *Pattern Recognition*, *43*(5), 1975–1983.

Kamyshanska, H., & Memisevic, R. (2015). The potential energy of an autoencoder. *IEEE transactions on pattern analysis and machine intelligence*, *37*(6), 1261–1273.

Kanamori, T., Hido, S., & Sugiyama, M. (2009). Efficient direct density ratio estimation for non-stationarity adaptation and outlier detection. In *Advances in neural information processing systems* (pp. 809–816).

Kim, K., & Lee, J. (2014a). Nonlinear dynamic projection for noise reduction of dispersed manifolds. *IEEE transactions on pattern analysis and machine intelligence*, *36*(11), 2303–2309.

Kim, K., & Lee, J. (2014b). Sentiment visualization and classification via semi-supervised nonlinear dimensionality reduction. *Pattern Recognition*, *47*(2), 758–768.

Kim, K., Son, Y., & Lee, J. (2015). Voronoi cell-based kernel support clustering. *IEEE Transactions on Knowledge & Data Engineering*(1), 1–1.

Kim, T., & Bengio, Y. (2016). Deep directed generative models with energy-based probability estimation. *arXiv preprint arXiv:1606.03439*.

Kingma, D. P., & Cun, Y. L. (2010). Regularized estimation of image statistics by score matching. In *Advances in neural information processing systems* (pp. 1126–1134).

Kloptchenko, A., Eklund, T., Karlsson, J., Back, B., Vanharanta, H., & Visa, A. (2004). Combining data and text mining techniques for analysing financial reports. *Intelligent systems in accounting, finance and management*, *12*(1), 29–41.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Lau, J. H., & Baldwin, T. (2016). An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv preprint arXiv:1607.05368*.

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In *Proceedings of the 31st international conference on machine learning (icml-14)* (pp. 1188–1196).

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., & Huang, F. (2006). A tutorial on energy-based learning. *Predicting structured data*, *1*, 0.

LeCun, Y., Cortes, C., & Burges, C. J. (2010). Mnist handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, *2*.

Lee, D., & Lee, J. (2007). Domain described support vector classifier for multi-classification problems. *Pattern Recognition*, *40*(1), 41–51.

Lee, D.-H. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, icml* (Vol. 3, p. 2).

Lee, J., & Chiang, H.-D. (2002). Theory of stability regions for a class of nonhyperbolic dynamical systems and its application to constraint satisfaction problems. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, *49*(2), 196–209.

Lee, J., & Lee, D. (2005). An improved cluster labeling method for support vector clustering. *IEEE Transactions on pattern analysis and machine intelligence*, *27*(3), 461–464.

Lee, J., & Lee, D. (2006). Dynamic characterization of cluster structures for robust and inductive support vector clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(11), 1869–1874.

Lin, C., & He, Y. (2009). Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th acm conference on information and knowledge management* (pp. 375–384).

Liu, H., Liu, T., Wu, J., Tao, D., & Fu, Y. (2015). Spectral ensemble clustering. In *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining* (pp. 715–724).

Liu, H., Shao, M., Li, S., & Fu, Y. (2016). Infinite ensemble for image clustering. In *Kdd* (pp. 1745–1754).

Long, M., Wang, J., Cao, Y., Sun, J., & Philip, S. Y. (2016). Deep learning of transferable representation for scalable domain adaptation. *IEEE Transactions on Knowledge and Data Engineering*, *28*(8), 2027–2040.

Maaten, L. v. d., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, *9*(Nov), 2579–2605.

Mcauliffe, J. D., & Blei, D. M. (2008). Supervised topic models. In *Advances in neural information processing systems* (pp. 121–128).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Mnih, A., & Kavukcuoglu, K. (2013). Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in neural information processing systems* (pp. 2265–2273).

Narayanan, H., Belkin, M., & Niyogi, P. (2007). On the relation between low density separation, spectral clustering and graph cuts. In *Advances in neural information processing systems* (pp. 1025–1032).

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., & Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning. In *Nips workshop on deep learning and unsupervised feature learning* (Vol. 2011, p. 5).

Nguyen, K. A., Walde, S. S. i., & Vu, N. T. (2016). Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction. *arXiv preprint arXiv:1605.07766*.

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, *22*(10), 1345–1359.

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, *2*(1–2), 1–135.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the acl-02 conference on empirical methods in natural language processing-volume 10* (pp. 79–86).

Park, E. (2016). *Supervised feature representations for document classification* (Unpublished doctoral dissertation). Seoul National University.

Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *2*(11), 559–572.

Perotte, A. J., Wood, F., Elhadad, N., & Bartlett, N. (2011). Hierarchically supervised latent dirichlet allocation. In *Advances in neural information processing systems* (pp. 2609–2617).

Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., & Lawrence, N. D. (2009). *Dataset shift in machine learning*. The MIT Press.

Rabhi, F. A., Guabtni, A., & Yao, L. (2009). A data model for processing financial market and news data. *International Journal of Electronic Finance*, *3*(4), 387–403.

Ramage, D., Hall, D., Nallapati, R., & Manning, C. D. (2009). Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 conference on empirical methods in natural language processing: Volume 1-volume 1* (pp. 248–256).

Ranzato, M., & Szummer, M. (2008). Semi-supervised learning of compact document representations with deep networks. In *Proceedings of the 25th international conference on machine learning* (pp. 792–799).

Rifai, S., Dauphin, Y. N., Vincent, P., Bengio, Y., & Muller, X. (2011). The manifold tangent classifier. In *Advances in neural information processing systems* (pp. 2294–2302).

Rifai, S., Vincent, P., Muller, X., Glorot, X., & Bengio, Y. (2011). Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th international conference on machine learning (icml-11)* (pp. 833–840).

Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *science*, *290*(5500), 2323–2326.

Scheible, S., Im Walde, S. S., & Springorum, S. (2013). Uncovering distributional differences between synonyms and antonyms in a word space model. In *Ijcnlp* (pp. 489–497).

Schölkopf, B., Smola, A., & Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, *10*(5), 1299–1319.

Seung, H. S. (1998). Learning continuous attractors in recurrent networks. In *Advances in neural information processing systems* (pp. 654–660).

Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, *90*(2), 227–244.

Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., & Manning, C. D. (2011). Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 151–161).

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., & Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing* (pp. 1631–1642).

Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, *28*(1), 11–21.

Strehl, A., & Ghosh, J. (2002). Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, *3*(Dec), 583–617.

Sugiyama, M., Krauledat, M., & MÃžller, K.-R. (2007). Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, *8*(May), 985–1005.

Sugiyama, M., Nakajima, S., Kashima, H., Buenau, P. V., & Kawanabe, M. (2008). Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in neural information processing systems* (pp. 1433–1440).

Swersky, K., Buchman, D., Freitas, N. D., Marlin, B. M., & Freitas, N. d. (2011). On autoencoders and score matching for energy based models. In *Proceedings of the 28th international conference on machine learning (icml-11)* (pp. 1201–1208).

Tated, R. R., & Ghonge, M. M. (2015). A survey on text mining-techniques and application. *International Journal of Research in Advent Technology*, *1*, 380–385.

Tax, D. M., & Duin, R. P. (1999). Support vector domain description. *Pattern recognition letters*, *20*(11), 1191–1199.

Tenenbaum, J. B., De Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *science*, *290*(5500), 2319–2323.

Torgerson, W. S. (1952). Multidimensional scaling: I. theory and method. *Psychometrika*, *17*(4), 401–419.

Vincent, P. (2011). A connection between score matching and denoising autoencoders. *Neural computation*, *23*(7), 1661–1674.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P.-A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, *11*(Dec), 3371–3408.

Weston, J., Ratle, F., Mobahi, H., & Collobert, R. (2012). Deep learning via semi-supervised embedding. In *Neural networks: Tricks of the trade* (pp. 639–655). Springer.

Xiao, M., & Guo, Y. (2015). Feature space independent semi-supervised domain adaptation via kernel matching. *IEEE transactions on pattern analysis and machine intelligence*, *37*(1), 54–66.

Yao, T., Pan, Y., Ngo, C.-W., Li, H., & Mei, T. (2015). Semi-supervised domain adaptation with subspace learning for visual recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 2142–2150).

Zellinger, W., Grubinger, T., Lughofer, E., Natschläger, T., & Saminger-Platz, S. (2017). Central moment discrepancy (cmd) for domain-invariant representation learning. *arXiv preprint arXiv:1702.08811*.

Zhai, S., & Zhang, Z. M. (2016). Semisupervised autoencoder for sentiment analysis. In *Aaai* (pp. 1394–1400).

Zhao, J., Mathieu, M., & LeCun, Y. (2016). Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*.

Zhong, C., Yue, X., Zhang, Z., & Lei, J. (2015). A clustering ensemble: Two-level-refined co-association matrix with path-based transformation. *Pattern Recognition*, *48*(8), 2699–2709.

Zinkevich, M., Weimer, M., Li, L., & Smola, A. J. (2010). Parallelized stochastic gradient descent. In *Advances in neural information processing systems* (pp. 2595–2603).

초록

데이터 관련 기술의 발전과 함께 점점 더 많은 원시 데이터가 생성되고 저장됨에 따라 데이터로부터 정보를 식별하는 것이 중요해지고 있다. 수집된 데이터를 분석하기 위해서 최근에는 기계학습과 딥러닝 모델들이 주로 사용되지만, 이러한 모델의 성능은 데이터 표현에 매우 의존적이다. 최근에 표현학습에 대한 연구들은 입력 밀도를 잘 파악하는 것이 데이터로부터 유용한 정보를 얻는데 도움이 된다는 것을 보여주었다. 이에 따라 본 연구는 밀도 기반 표현 학습에 중점을 두었다. 고차원의 데이터의 경우에 실제로 더 낮은 차원의 높은 밀도의 영역에 데이터가 집중되어 있기 때문에 다양체(manifold) 가정은 표현학습에서 중요한 개념 중에 하나이다. 또한 비정형 데이터는 기계학습이나 딥러닝 모델들을 적용하기 전에 수치 벡터로 변환되어야 한다. 텍스트 데이터의 경우에는 분포 기반 표현 학습(distributed representation)이 단어와 문서에 대한 연속적인 벡터를 구하면서도 효과적으로 데이터의 정보들을 반영할 수 있었다. 본 연구에서는 분포 기반 표현 학습의 관점에서 입력 데이터의 다양체나 텍스트 데이터의 분포 기반 표현에 관한 몇 가지 문제들을 해결하였다.

먼저, 입력 데이터의 밀도가 다양체 상의 분포(distribution on manifold)로 한정되었을 경우에 동적 시스템의 관점에서 denoising autoencoder(DAE)를 살펴보았다. 가우시안 분포가 합성된 입력 데이터를 이용해 학습된 DAE로부터 입력 데이터의 score 함수를 추정하여 동적 사영 시스템(dynamical projection system)을 구축하였다. 몇 가지 분석을 통해 유도된 동적 시스템의 안정성(stability)과 입력 데이터의 다양체로의 수렴성(convergence)을 입증하였고 이것으로부터 비선형 사영 알고리듬(a nonlinear projection algorithm)을 개발하여 고밀도 영역(high-density region)을 찾거나 노이즈가 낀 입력 데이터의 노이즈를 감소하는데 활용하였다. 또한, 본 알고리듬의 효과는 몇 개의 toy examples과 실제 이미지 데이터 셋들을 통해서 입증하였다.

Support vector domain description(SVDD) 모형은 margin과 커널 파라미터에 대한

약한 조건 하에서 학습된 kernel radius function을 통해서 입력 밀도를 추정할 수 있다. 이러한 사실로부터 여러 군집화결과들로부터 새로운 유사도를 정의하여 kernel support matching을 통해서 kernel radius function을 구하는 새로운 inductive ensemble 군집화 방법론을 개발하였다. 실험을 통해 제안된 방법이 효과적일 뿐만 아니라 robust한 군집화 결과를 보여줌을 확인하였다. DAE 모형은 데이터 서포트 밖의 영역에 대한 제약을 하지 않기 때문에 고밀도 영역 사이의 저 밀도 영역에 관해 잘못된 에너지를 학습할 수 있는 반면 SVDD 모형은 kernel radius function을 학습함으로써 입력 데이터의 서포트 영역을 잘 잡아준다. 따라서, 이러한 성질을 활용하여 DAE를 학습할 때 저밀도 영역에 대해 학습된 kernel radius function을 통해 regularization을 해 줄 수 있는 방법을 개발하였다. 예제를 통해서 이러한 regularization의 효과를 살펴보았다.

문서의 표현을 학습하는 것은 감성 분석에 기계 학습 방법론들을 활용하기 위해서 중요하다. 단어와 문서의 분포 기반 표현 학습 방법은 자연어 처리에서 성공적으로 활용되었지만, 이러한 모형들이 단어 사이의 문맥을 기반으로 한 목적함수만 가지고 학습하기 때문에 학습된 표현이 문서의 감성 정보를 반영하지 못한다. 따라서, 일부의 문서에 감성 정보가 주어진 경우에 반교사 감성 분류(semi-supervised sentiment-discriminative) 목적함수를 추가한 새로운 분포 기반 문서 학습 방법론을 제안하였다. 제안된 방법론은 근접한 문서들의 감성 정보만 반영함으로써 감성 정보를 반영하면서도 국소 구조를 보존하도록 하였다. 실제 문서 데이터를 가지고 시각화와 예측 분석을 수행한 결과 제안된 방법은 다른 표현 방법론들보다 우수한 성능을 보였다.

또한 문서들은 도메인에 따라서 매우 다른 특성을 가질 수 있으므로 도메인 적응(domain adaptation)에서 자연어 처리는 중요한 응용 분야 중에 하나이다. 자연어 처리에 관한 많은 도메인 적응 방법들은 직접 텍스트 입력으로부터 모델을 학습하기보다 수치 벡터들로 부터 공통 특성을 학습한다. 따라서 본 연구에서는 분포 기반 문서 표현 방법이 서로 다른 도메인들로부터의 문서들이 함께 학습될 때에 분포 차이에 의해 학습된 벡터의 서포트가 분리될 수 있는 문제를 해결하는 도메인 적응 분포 기반 표한 학습 방법론을 개발하였다. 이러한 방법은 도메인에 대해 분리되지 않는 문서 표현을 학습하면서도 텍스트 입력으로부터 직접 수치 벡터를 구하기 때문에 다른 도메인 적응

방법론들의 기본 수치 표현으로 활용될 수 있다. 시각화와 감성 분석에 대한 실험을 통해 제안된 방법론이 일관적으로 좋은 결과를 줄 수 있음을 확인하였다.

최근에 고차원의 표현을 가지거나 텍스트의 형태로 존재하는 많은 데이터들이 있기 때문에 고차원의 데이터에서 다양체 구조를 파악하고 유용한 정보들을 반영하는 텍스트 데이터의 수치 표현을 구하는 표현 학습 방법론이 필요하다. 따라서 우리의 방법론들이 이러한 요구를 충족시키는데 도움이 되고 다양한 데이터 분석에 활용되기를 기대할 수 있다.

주요어: 표현 학습, 다양체 학습, 잡음제거 자동부호화기, 분포 기반 표현, 감성 분석, 도메인 적응

학번: 2013-21068